МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра информационных систем управления

МЕНЬКОВ АНДРЕЙ АЛЕКСАНДРОВИЧ

РАЗРАБОТКА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОБНАРУЖЕНИЯ АТАК НА КОМПЬЮТЕРНЫЕ СИСТЕМЫ

Отчёт по преддипломной практике
студента 5 курса 2 группы

Руководитель:
Образцов Владимир Алексеевич
доцент кафедры ИСУ,
кандидат физико-математических наук

МИНСК
БГУ
2013

АННОТАЦИЯ

Меньков А. А. Разработка математического и программного обеспечения для решения задачи обнаружения атак на компьютерные системы: Отчёт по преддипломной практике / Минск: БГУ, 2013. — 10 с.

АНАТАЦЫЯ

Менькоу А. А. Распрацоўка матэматычнага і праграмнага забеспячэння для вырашэння задачы выяўлення нападаў на кампутарныя сістэмы: Справаздача аб пераддыпломнай практыцы / Мінск: БДУ, 2013. — 10 с.

ANNOTATION

Menkou A. A. Development of mathematical and software solutions for the problem of detection of attacks on computer systems: Pregraduation report / Minsk: BSU, 2013 — 10 p.

РЕФЕРАТ

Отчёт по преддипломной практике, 10 с., ? рис., ? источников.

Ключевые слова: КОМПЬЮТЕРНАЯ СЕТЬ, АТАКА, ОБНАРУЖЕНИЕ АТАК.

Объект исследования — атаки на компьютерные сети.

Цель работы — разработать обеспечение для обнаружения атак на компьютерные сети.

Методы исследования — методы прикладной математики и информатики, технология программирования.

Результат исследования — .

Областью применения являются компьютерные сети с повышенным контролем безопасности информации, передаваемой по сети.

СОДЕРЖАНИЕ

Список иллюстраций

ВВЕДЕНИЕ

Компьютерные сети за несколько последних десятилетий из чисто технического решения превратились в глобальное явление, развитие которого оказывает влияние на большинство сфер экономической деятельности. Одним из первых количественную оценку значимости сетей дал Роберт Меткалф, участвовавший в создании Ethernet: по его оценке «значимость» сети во всех смыслах пропорциональна квадрату числа узлов в ней. То есть, зависимость от нормальной работы сетей растёт быстрее, чем сами сети. Обеспечение работоспособности сети и функционирующих в ней информационных систем зависит не только от надёжности аппаратуры, но и, зачастую, от способности сети противостоять целенаправленным воздействиям, которые направлены на нарушение её работы.

Создание информационных систем, гарантированно устойчивых к вредоносным воздействиям и компьютерным атакам, сопряжено с существенными затратами как времени, так и материальных ресурсов. Кроме того, существует известная обратная зависимость между удобством пользования системой и её защищённостью: чем совершеннее системы защиты, тем сложнее пользоваться основным функционалом информационной системы. В 80-е годы XX века, в рамках оборонных проектов США, предпринимались попытки создания распределенных информационных систем специального назначения (MMS – Military Messaging System) [103], для которых формально доказывалась выполнимость основной теоремы безопасности – невыведение системы из безопасного состояния для любой последовательности действий взаимодействующих объектов. В этих системах использовалось специализированное программное обеспечение на всех уровнях, включая системный. Однако, на сегодняшний день подобные системы не получили развития, и для организации информационных систем используются операционные системы общего назначения, такие как ОС семейства Microsoft Windows, GNU/Linux, *BSD и различные клоны SysV UNIX (Solaris, HP-UX, etc).

Методы обнаружения атак в современных системах обнаружения атак (далее - СОА) недостаточно проработаны в части формальной модели атаки, и, следовательно, для них достаточно сложно строго оценить такие свойства как вычислительная сложность, корректность, завершимость. Принято разделять методы обнаружения атак на методы обнаружения аномалий и методы обнаружения злоупотреблений. Ко второму типу методов относятся большинство современных коммерческих систем (Cisco IPS, ISS RealSecure, NFR) — они используют сигнатурные (экспертные) методы обнаружения. Для таких систем основной проблемой является низкая, близкая к нулю, эффективность обнаружения неизвестных атак (адаптивность). Низкая адаптивность до сих пор остаётся проблемой, хотя такие достоинства как низкая вычислительная сложность и малая стоимость развёртывания определяют доминирование таких систем в данной области.

1 ОБЗОР СУЩЕСТВУЮЩИХ СТАНДАРТОВ ШТРИХКОДОВ

Штрихкод (сокр. от «*штриховой код*», англ. «*bar code*») — графическая информация наносимая на поверхности предметов, предназначенная для обработки техническими средствами.

Выделяют две большие группы штрихкодов: *линейные* и *двухмерные*. В первом случае информационную нагрузку имеют только чередования участков различной яркости по одной из осей, во втором — по обеим.

1.1 Линейный штрихкоды

Исторически, линейные штрихкоды появились первыми (50–70-ые годы XX века). Коды этой группы не отличаются особым разнообразием — чередование чёрных и белых полос, кодирующие цифры либо буквы. Вместе с тем, линейные коды наносятся практически на все товары (Рисунок 1), распространяемые в розницу (формат EAN — European Article Number), и потому наиболее широко распространены. Другие примеры рисунки 2, 3.

В этой работе мы не будем рассматривать линейные штрихкоды.

Рисунок 1 – Штрихкод в формате EAN

Рисунок 2 – Штрихкод в формате Code 128

Рисунок 3 – Штрихкод в формате ITF-14

1.2 Двумерные штрихкоды

Как уже отмечалось, в двумерных штрихкодах информация кодируется по обеим осям. Выделяются два вида двумерных кодов: *стековые* и *матричные*.

1.2.1 Стековые штрихкоды

Стековые штрикоды являются, в некоторой мере, переходным вариантом между линейными и двухмерными. Они представляю собой строки расположенные одна под одной. Каждая строка есть ни что иное, как одномерный штрихкод.

Несмотря на свою простоту, данный тип штрихкодов позволяет хранить значительные объёмы данных. Например, штрикод формата PDF417 (см. Рисунок 4) может содержать до 2710 знаков. Стековые штрихкоды также не является объектом рассмотрения данной работы.

Рисунок 4 – Штрихкод в формате PDF417

1.2.2 QR-код

Ниже будет приведено краткое описание QR-кода. Полное описание можно найти в спецификации ISO/IEC 18004:2006 [**?**].

QR-код представляет собой квадратную матрицу размером от 21×21 до 170×170 барселей. Технически выделяют от 1 (21×21) до 40 (170×170) версии кода (т.е. шаг равен 4).

Каждый барсель есть некоторый аналог бита при представлении информации в электронном виде. Среди этих барселей можно выделить следующие (Рисунок 5): *информационные* — содержат непосредственно данные, *корректирующие* — для исправления ошибок (используются коды Рида-Соломона (см. 2.1)), *заполняющие* — дополняют до правильно квадрата, *поисковые шаблоны* — служат для локации кода при распознавании (хорошо заметные квадраты по углам кода, а также чередующие чёрные и белые барсели чуть ниже них), *форматные* — содержат информацию о структуре кода (в QR-коде расположены по периметру поисковых шаблонов). Дополнительно смотрите Рисунок 6

Рисунок 5 – Штрихкод в формате QR-код

В одном коде можно сохранить до 7084 цифр, либо до 4296 цифр и символов, либо до 2953 байт произвольных данных, либо до 1817 иероглифов Канджи.

Дополнительно фиксируется уровень коррекции ошибок *L*, *M*, *Q*, *H*, что позволяет восстановить до 7%, 15%, 25%, 30% данных соответственно. Рекомендуется использовать уровень *M*.

При распечатке требуется обеспечить белые поля толщиной не менее четырёх барселей.

1.2.3 Data Matrix

Data Matrix — двумерный штрихкод (описывается стандартом ГОСТ Р ИСО/МЭК 16022–2008, аналогом ISO/IEC 16022:2006 [**?**]), позволяющий закодировать до 2335 алфавитно-цифровых символов, либо до 3116 цифр, либо до 1556 байтов информации (см. Рисунок 7). Data Matrix, как и все другие подобные штрихкоды, содержит информацию для восстановления, которая позволяет восстановить закодированную информацию при частичном повреждении кода.

Каждый код Data Matrix содержит две сплошные пересекающиеся линии в виде буквы L, для ориентации считывающего устройства, две другие границы кода состоят из перемежающихся чёрных и белых точек и служат для указания размеров кода считывающему

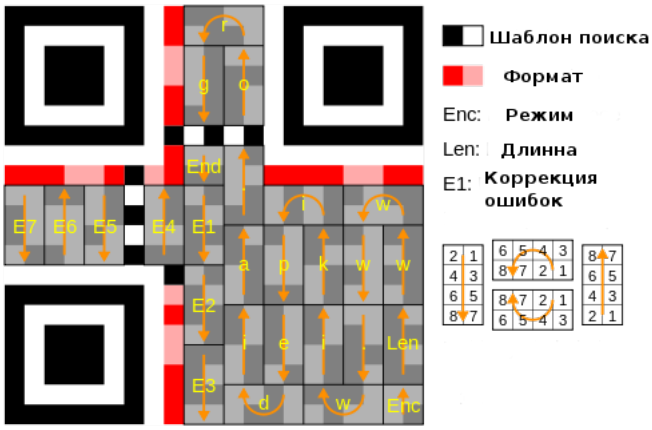

Рисунок 6 – Пример декодирования QR-кода

Рисунок 7 – Штрихкод в формате Data Matrix

устройству (см. Рисунок 8). Размер кода может быть от 10×10 до 144×144 барселей (существую также прямоугольные версии для цилиндрических поверхностей). Дополнительно можно объединять до 16 кодов в один большой символ. На рисунке 9 показано, как размещаются байты в штрихкоде.

Рисунок 8 – Шаблон поиска в штрихкоде Data Matrix

Все коды используют коррекцию ошибок стандарта ECC200, который, в свою очередь, использует алгоритм Рида-Соломона для кодирования/декодирования данных. Это позволяет восстановить в случае повреждения кода до 30% полезной информации.

В промышленности Data Matrix применяют для маркировки различных элементов. Код может быть нанесён различными способами — струйной печатью, гравировкой, лазером, электролитическими способами и т.д. В зависимости от метода нанесения, код может оставаться на элементе на протяжении всего его цикла использования (Рисунок 10).

Основным положительным отличием Data Matrix от остальных двухмерных штрихкодов является то, что Data Matrix работает максимально быстро и небольшие объёмы данных осуществляются на минимальных площадях. Так, если кодировать 6 цифр, Data Matrix составит штрих-код размером 10×10 модулей, а вот если кодировать в Aztec, то эта площадь будет составлять 15×15 модулей. Но в то же время, данное преимущество теряется в процессе кодирования большего объёма информации: при одинаковых размерах символа в 132×132 модуля, Aztec закодирует почти 3000 цифр, а Data Matrix максимум 2608. И при кодирова-

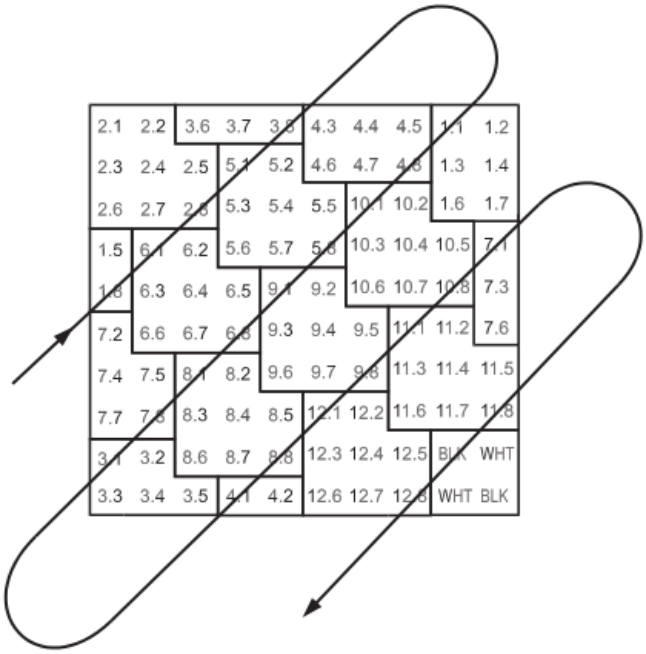

Рисунок 9 – Расположение байтов в Data Matrix

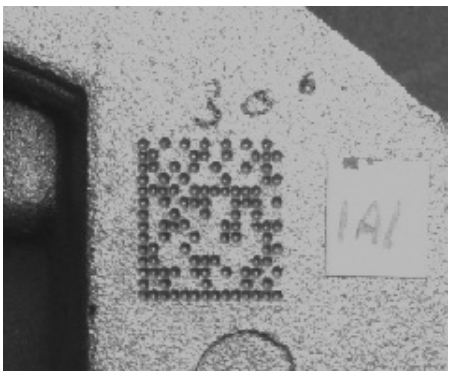

Рисунок 10 – Штрихкод Data Matrix нанесённый методом иглографии

нии буквенно-цифровых данных в 10 символов, штрих-код Data Matrix занимает одинаковую площадь с Aztec. Чем же объясняется столь успешное кодирование малых пользовательских данных? Прежде всего, малым количеством служебной информации в составляемом коде, экономия производится за счёт размеров и структуры данных штрих-кода, таким образом надёжность считывания несколько падает. Также, Data Matrix имеет ещё один недостаток — рост размера шаблона поиска символов увеличивается прямо пропорционально самому периметру символа, таким образом, становится невыгодным кодирование больших объёмов данных, у Aztec же шаблон поиска не изменяется.

1.2.4 Aztec Code

Aztec Code представляет собой универсальную символику двухмерного штрихового кода. Как показано на рисунках 11 и 12, код представляет собой квадрат, содержащий матрицу квадратных элементов, в центре которой располагается «мишень» («bullseye[1]»), составленная из концентрических квадратов. Aztec позволяет эффективно кодировать как малые, так и большие объемы данных (цифры — до 3832, текст — до 3067 или байты — до 1914) с использованием высокоэффективного метода Рида-Соломона коррекции ошибок (см. 2.1). Код

[1]Т.е. «бычий глаз»

Таблица 1 – Таблица возможностей Data Matrix

Размер	Количество кодируемой информации		
Ширина × Высота	Шифры	Символы	Байты
10×10	6	3	1
12×12	10	6	3
14×14	16	10	6
16×16	24	16	10
18×18	36	25	16
20×20	44	31	20
22×22	60	43	28
24×24	72	52	34
26×26	88	64	42
32×32	124	91	60
40×40	228	169	112
44×44	288	214	142
48×48	348	259	172
52×52	408	304	202
64×64	560	418	278
72×72	736	550	366
80×80	912	682	454
88×88	1152	862	574
96×96	1392	1042	694
104×104	1632	1222	814
120×120	2100	1573	1048
132×132	2608	1954	1302
144×144	3116	2335	1556
8×18	10	6	3
8×32	20	13	8
12×26	32	22	14
12×36	44	31	20
16×36	64	46	30
16×48	98	72	47

Aztec разработан специалистами фирмы HandHeld Products (Andy Longacre и Rob Hussey) и защищен патентом, но частично выпущен для общего использования. Международная Спецификация Символики для кода Aztec утверждена AIM USA в формате ISO и доступна через филиалы AIM.

Квадратная «мишень», окружённая «слоями данных», сплетенными с решеткой «элементов привязк», расположенной по периметру квадрата, дают в результате изображения ассоциирующееся с искусством Центральной Америки, что и подсказало имя «Aztec Code» для символики.

Основные изменения в структуре кода и коррекции ошибок появились в Версии 2.0 спецификации в июне 1995 года, но основная конструкция кода осталась неизменной, выдержав процесс отладки считывающих устройств, пробные внедрения и даже критический анализ, проведённый Техническим Комитетом (Technical Symbology Committee) AIM USA без изменений. Международная спецификация Aztec Code опубликована AIM International в 1997 году.

Существуют два основных формата символа Aztec Code: «Compact» (Компактный) сим-

Рисунок 11 – «Компактный» символ Axtec Code

Рисунок 12 – «Полноразмерный» символ Axtec Code

вол с мишенью из двух квадратов (Рисунок 11) и «Full-Range» (Полный) символ с мишенью из трёх квадратов (Рисунок 12). Поскольку принтеры могут автоматически выбирать, а сканеры автоматически распознавать оба формата, вместе два формата образуют последовательность из символов 33 различных размеров, которые могут эффективно кодировать как малые, так и большие сообщения. В общем, символы Aztec Code:

1. могут кодировать любую байтовую последовательность в эффективных компактных режимах для текстовых и цифровых данных;
2. всегда квадратной формы, изменяясь в размерах от 15×15 модулей до 151×151 модулей. Свободной зоны вокруг символа не требуется вообще. Таблица 2 показывает информационную ёмкость некоторых размеров кода;
3. может быть использован в структурном объединении, соединяющем до 26 символов;
4. имеет специальный формат настройки сканера, удобный для настройки сканера с помощью штрихкода.

Таблица 2 – Соотношения размеров символов и ёмкости Aztec Code

Вид символа Aztec Code очень систематичен с чётко разграниченными функциями частей, обеспечивает простоту процедур кодирования и декодирования, в то же время его математическая структура необычайно гибка и надёжна.

Рисунок 13 показывает структуру полного символа Aztec Code. Вы можете увидеть три постоянных элемента:

1. центральный указатель «мишень»;
2. элементы ориентации по углам указателя;
3. решётка привязки, пронизывающая область данных.

Два переменных элемента структуры[2]

1. строка короткого режима, обернутая вокруг «мишени»;
2. от одного до 32 слоев данных толщиной в 2 барселя, спиралью расходящихся от центра.

[2]Компактный символ Aztec Code содержит маленькую мишень без решётки привязки и только 4 слоя данных.

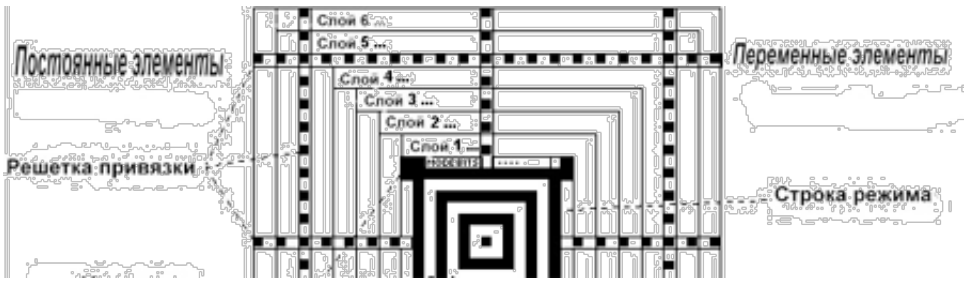

Рисунок 13 – Структура Aztec Code

Строка короткого режима и слои данных закодированы с защитой от ошибок по методу Рида-Соломона. Строка режима — это строка фиксированной длины, которая просто кодирует два параметра, несущие информацию о слоях данных, а именно: сколько слоев данных содержит данный символ и сколько слов содержится в сообщении (остаток места в области данных заполняется контрольными словами). Таким образом, уровень коррекции ошибок в Aztec Code становится регулируемым по указанию пользователя, и в принципе, слои данных могут содержать от 5% до 95 % контрольных слов, но на практике обычно нецелесообразно изменять стандартное значение в 23% контрольных слов.

Слои данных, конечно, содержат последовательность кодовых слов, которые сперва кодируют пользовательские данные, затем добавляют к ним выявление и коррекцию ошибок. Защита от ошибок, кроме того, регулируемая пользователем и использующая дополнительные контрольные слова для заполнения, дополнительно усилена двумя путями: во-первых, размер кодового слова зависит от размера символа, от 6 бит для наименьших символов до 12 бит для наибольших, исключая необходимость чередующихся полей и обеспечивая хорошую зернистость для всех размеров символов. Во-вторых, слова сообщения, занимающие внешние слои символа, поддерживают чистовую коррекцию ошибок в стёртых углах символа.

В результате представленного рассмотрения технологии становятся понятными некоторые особенности Aztec Code:

1. Слоёная природа полей данных обеспечивает целостность символов 33 различных размеров и информационной ёмкости.

2. Указатель в виде мишени обеспечивает считывание при большом изменении угла сканирования.

3. Элементы ориентации дают возможность считывания при любой ориентации символа, включая зеркальное отражение.

4. Решётка привязки позволяет учитывать существенные искривления больших символов.

5. Декодирование от центра к краю исключает необходимость полей (свободной зоны) вокруг символа.

6. Надёжный управляемый пользователем механизм коррекции ошибок по методу Рида-Соломона обеспечивает высокую производительность и надёжную защиту от ошибок.

7. Расположение полей, устойчивых к появлению ошибок и повреждений, по краям символа, компенсирует влияние оптических искажений, возникающих по краям зоны сканирования.

Aztec Code представляет собой универсальную символику матричного штрихового кода, хорошо приспособленную для визуальной технологии считывания и для кодирования как малых, так и больших объёмов данных. Aztec Code интересен для применений, требующих

размещения кода на ограниченном пространстве (производство, коммерция, медицина, фармацевтика и т.д.), поскольку код обеспечивает высокую плотность размещения информации и не требует свободного пространства вокруг кода. Некоторые почтовые ведомства рассматривают возможность использования Aztec Code в качестве «электронного штампа» почтового отправления, в то же время электронное кодирование подписи с помощью Aztec привлекло внимание некоторых транспортных компаний.

1.3 Эффективность кода

Эффективностью штрихкода C назовём величину

$$\theta_C(b, q) = \frac{b}{d(b, q)}, \tag{1}$$

где $d(b, q)$ — число барселей, которые необходимы для того, чтобы зашифровать сообщение длинной b бит при конфигурации кода q (отражает уровень коррекции ошибок и т.д.).

Другими словами, эффективность кода отражает число бит полезной информации которое несёт в себе каждый барсель рассматриваемого кода C.

Предельной эффективностью кода назовём величину

$$\Theta_C(q) = \lim_{b\to\infty} \theta_C(b, q). \tag{2}$$

Для рассмотренных выше кодов имеет место (принимаем во внимание, что коды чёрно-белые, а значит один барсель — один бит):

$$d(b, q) = b + 2\lceil bg(q)\rceil + f(b, q) + s(b, q),$$

где $g(q)$ — текущий уровень коррекции ошибок (везде используется коды Рида-Соломона, поэтому на исправление одной ошибки требуется два дополнительных символа (см. 2.1)), $f(b, q)$ — свободное место в матрице, $s(b, q)$ — служебная информация в коде[3]. Принимая во внимание тот факт, что $f(b, q) = o(b)$, а также $f(b, q) = o(b)$ (действительно, в рассматриваемых кодах информация записывается таким образом, что свободное место минимизируется; а служебные данные с возрастанием информации практически не возрастают) имеем[4]

$$\Theta_C(q) = \lim_{b\to\infty} \frac{b}{b + 2\lceil bg(q)\rceil + o(b)} = \frac{1}{1 + 2g(q)}. \tag{4}$$

[3]В случае, когда каждый барсель может находиться в более чем двух состояниях k будем иметь:

$$d(b, q) = \frac{b + 2\lceil bg(q)\rceil}{\log_2 k} + f(b, q) + s(b, q),$$

[4]Соответственно, для случая, описанного в предыдущем примечании:

$$\Theta_C(q) = \frac{\log_2 k}{1 + 2g(q)}. \tag{3}$$

2 ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ И РАСПОЗНАВАНИЯ ДВУМЕРНЫХ ШТРИХКОДОВ

2.1 Коды Рида-Соломона

2.2 Перспективные преобразования

2.3 Общий алгоритм распознавания двумерных штрихкодов

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] *Лопатин В. В. и др.* Русский орфографический словарь: около 180 000 слов / Иванова О. Е., Лопатин В. В., Нечаева И. В., Чельцова Л. К. Отв. ред. В. В. Лопатин. — 2-е изд., испр. и доп. — М.: Институт русского языка имени В. В. Виноградова РАН, 2004. — 960 с.