В. А. ГРИДНЕВ, Ю. А. ГУБСКОВ,
А. С. ДЕРЯБИН, А. В. ЯКОВЛЕВ

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

В ТРЕХ ЧАСТЯХ

ЧАСТЬ 3

Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2024

Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тамбовский государственный технический университет»**

**В. А. ГРИДНЕВ, Ю. А. ГУБСКОВ,
А. С. ДЕРЯБИН, А. В. ЯКОВЛЕВ**

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

В ТРЕХ ЧАСТЯХ

ЧАСТЬ 3

Утверждено Ученым советом университета в качестве
учебного пособия для студентов 5 курса специальности
10.05.03 «Информационная безопасность автоматизированных систем»
очной формы обучения и 3 курса направления подготовки 09.03.02
«Информационные системы и технологии»
очной и заочной форм обучения

Учебное электронное издание

Тамбов
Издательский центр ФГБОУ ВО «ТГТУ»
2024

УДК 004.065
ББК 32.973.26
П78

Р е ц е н з е н т ы :

Кандидат технических наук, доцент, преподаватель цикла «Применение комплексов РЭБ и средств комплексного технического контроля» ФКУ в/ч 61460
К. А. Малыков

Директор Центрально-Черноземного РУНЦ ИБ ФГБОУ ВО «ТГТУ»
П. А. Щербинин

П78 **Программно-аппаратные средства** защиты информации [Электронный ресурс] : учебное пособие : в 3-х ч. / В. А. Гриднев, Ю. А. Губсков, А. С. Дерябин, А. В. Яковлев. – Тамбов : Издательский центр ФГБОУ ВО «ТГТУ».
ISBN 978-5-8265-2464-0

Ч. 3. – 2024. – 1 электрон. опт. диск (CD-ROM). – Системные требования : ПК не ниже класса Pentium II ; CD-ROM-дисковод ; 1,7 Mb ; RAM ; Windows 95/98/XP ; мышь. – Загл. с экрана.
ISBN 978-5-8265-2795-5

Посвящено вопросам построения и функционирования программно-аппаратных средств защиты информации в автоматизированных системах, являющихся основой комплексной системы информационной безопасности организации. Включает в себя полный теоретический курс учебной дисциплины «Программно-аппаратные средства обеспечения информационной безопасности».

Предназначено для студентов 5 курса специальности 10.05.03 «Информационная безопасность автоматизированных систем» очной формы обучения и 3 курса направления подготовки 09.03.02 «Информационные системы и технологии» очной и заочной форм обучения.

УДК 004.065
ББК 32.973.26

ISBN 978-5-8265-2464-0 (общ.)
ISBN 978-5-8265-2795-5 (ч. 3)

ВВЕДЕНИЕ

Учебное пособие состоит из трех частей и включает в себя полный теоретический курс учебной дисциплины «Программно-аппаратные средства защиты информации», преподаваемой в ФГБОУ ВО «Тамбовский государственный технический университет» студентам, обучающимся по специальности 10.05.03 «Информационная безопасность автоматизированных систем».

Данное учебное пособие способствует привитию студентам общепрофессиональной компетенции ОПК-15 (способен осуществлять администрирование и контроль функционирования средств и систем защиты информации автоматизированных систем, инструментальный мониторинг защищенности автоматизированных систем) [1].

Учебное пособие может быть полезно также студентам, обучающимся по направлению подготовки бакалавров 09.03.02 «Информационные системы и технологии» при изучении дисциплины «Информационная безопасность и защита информации» для формирования общепрофессиональной компетенции ОПК-3 (способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности).

Часть 3 пособия включает разделы: защита ЭВМ от вредоносного программного обеспечения; технологическая безопасность информационных систем и системы обнаружения вторжений.

1. ЗАЩИТА ЭВМ ОТ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1. КЛАССИФИКАЦИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В XXI веке почти все люди имеют представление о компьютерных вирусах, даже если они непосредственно не связаны с компьютерами по роду своей деятельности. Но далеко не каждый из них догадывается, что компьютерные вирусы – всего лишь часть вредоносного программного обеспечения (ВПО), причем, не самая большая и не самая опасная. С одной стороны, далеко не каждая, потенциально опасная для компьютера программа, является вирусом, а с другой – не все вирусы представляют опасность, среди них есть и совершенно безвредные экземпляры. Чтобы получить представление о всем многообразии ВПО, дадим его классификацию, т.е. разделим его на классы и виды.

При этом, надо иметь в виду, что единой классификации ВПО, подобной классификации химических элементов по периодической таблице Д. И. Менделеева, не существует. Поэтому специалисты каждой из компаний, которые занимаются защитой компьютеров от ВПО, используют свою классификацию, в соответствии с которой они относят вредоносный программный код к тому или иному виду или классу. Поэтому один и тот же вредоносный код различными специалистами может называться по-разному. Далее в настоящем пособии будет использована классификация ВПО, которой придерживается лаборатория Касперского. Согласно этой классификации, вредоносное ПО подразделяется на четыре вида, которые, в свою очередь, делятся на классы.

1.1.1. СЕТЕВЫЕ ЧЕРВИ

Сразу заметим, что сетевые черви еще в конце прошлого века уже потеряли свою актуальность, потому что они перестали пользоваться популярностью среди создателей вредоносных программ. Да и назы-

вать настоящими создателями ВПО представителей этого «движения» можно лишь условно, так как большинство из них являются школьниками или студентами, которые вовсе не являются программистами, а лишь используют конструкторы вредоносных программ. Случаи создания сетевых червей, которые действительно представляли бы опасность для информационных систем, встречаются крайне редко. Так, что же такое сетевой червь?

Сетевой червь – это вид ВПО, обладающего свойствами саморазмножения и распространения своих копий по компьютерным сетям.

Для самораспространения сетевые черви могут использовать:

- файлообменные сети;
- сети интернет-реал-чатов;
- сети мессенджеров;
- электронную почту;
- сети обмена данными между мобильными устройствами;
- локальные сети.

Именно способ самораспространения и положен лабораторией Касперского в основу разделения сетевых червей на классы.

Пути, по которым сетевые черви могут проникать на компьютер-жертву, могут быть различными, но их можно свести в три группы:

- самостоятельный путь;
- пользовательский путь;
- использование уязвимостей в системах безопасности ПО.

Черви, использующие самостоятельный способ распространения, называются пакетными. Пользовательский способ распространения называют также социальным инжинирингом.

Обычно сетевые черви распространяются в виде файлов, но возможно использование ссылки на файл или вложения в электронное письмо. Но есть и разновидность червей, которые могут распространяться в виде сетевых пакетов. Именно этот класс червей считают наиболее опасным, так как их распространение не зависит от пользователя.

После внедрения на компьютер-жертву черви попадают в память ЭВМ и становятся резидентами – создают свою копию на этом компьютере, запускают ее на исполнение и продолжают дальнейшее распространение.

Этот вид ВПО может содержать экземпляры, обладающие свойствами вредоносных программ других видов (обычно троянских). Рассмотрим подробнее некоторые классы сетевых червей.

Почтовые черви. Сетевые черви, которые используют для самораспространения электронную почту, относятся к классу почтовых или ***Email-Worm*.** Обычно черви данного класса отправляют компьютеру-жертве электронные письма, содержащие вредоносный код, но иногда письмо может содержать не сам вредоносный код, а ссылку на ресурс, содержащий вредоносный код. Для отправки сообщений почтовые черви могут использовать:

– сервисы почтового клиента;

– прямое подключение к почтовому серверу, с использованием встроенной в код червя специальной почтовой библиотеки;

– функции почтового сервиса, интегрированного в ОС.

Для поиска адресов электронной почты *Email-Worm* могут сканировать файлы, выделяя те, которые содержат адреса электронной почты по характерным для них признакам (например, наличие символа @) или использовать адресную книгу почтового клиента, установленного на зараженном компьютере. По всем обнаруженным в почтовом ящике адресам ВПО данного вида может рассылать свои копии. Существуют экземпляры почтовых червей, которые могут даже отвечать на письма из почтового ящика, а отдельные представители червей этого класса способны комбинировать способы поиска жертв и самораспространения.

Черви, использующие интернет-пейджеры. Сетевые черви этого класса (*IM-Worm*), как можно понять уже из их названия, используют для самораспространения контакты, обнаруженные на сервере интернет-пейджинга, что практически полностью повторяет способ распространения почтовых червей. С прекращением применения интернет-пейджеров, также утратил свою популярностьи данный класс сетевых червей.

Черви, использующие каналы интернет-реал-чатов. Сетевые черви этого класса (*IRC-Worm*) распространяют свои копии двумя путями – прямым и косвенным. Прямой путь предполагает отправку непосредственно зараженного файла, а косвенный использует ссылку на интернет-ресурс, содержащий зараженный файл. При этом во всех

случаях пользователь обязательно должен подтвердить прием этого файла.

Черви для файлообменных сетей. Сетевые черви данного класса для самораспространения используют создание собственных копий и размещение их в каталоге файлообменной сети *P2P.* Теперь уже сама файлообменная сеть будет давать положительные ответы на поисковые запросы пользователей, которые осуществляют поиск нужного им файла, и предоставлять все необходимое для его скачивания.

Более сложные черви данного класса сами имитируют сетевой протокол конкретной файлообменной системы, положительно отвечают на поисковые запросы, но для скачивания предлагают собственную копию.

Прочие сетевые черви. Существующие разновидности сетевых червей не ограничиваются перечисленными классами. Существует и много других разновидностей ВПО данного вида. Их принято классифицировать по способу распространения. Вот еще некоторые примеры способов распространения сетевых червей:

- копирование на сетевые ресурсы;
- использование брешей в ОС и прикладном ПО;
- через публично используемые сетевые ресурсы;
- с помощью других вредоносных программ.

Для прочих сетевых червей существует общее название ***NET-Worm***.

1.1.2. КЛАССИЧЕСКИЕ ВИРУСЫ

Определение термина «классический вирус» похоже на определение предыдущего класса ВПО – сетевых червей. Вирусы, также как и сетевые черви, обладают свойствами саморазмножения и самораспространения. Но, в отличие от сетевых червей, классические вирусы не используют для своего распространения сетевые сервисы и способ их попадания на компьютер-жертву не заложен в сам его программный код. Чаще всего распространение классических вирусов происходит путем социального инжиниринга, когда пользователь не применяет антивирусные средства для проверки файлов, записываемых на свой компьютер. Перечислим некоторые пути внедрения классических вирусов на компьютер-жертву:

– запись файлов с внешних носителей информации;

– скачивание файлов с интернет-ресурсов;

– скачивание файлов, распространяющихся по локальным сетям.

Классические вирусы, также как и сетевые черви, могут обладать свойствами других типов вредоносных программ (обычно троянских). По классификации лаборатории Касперского классические вирусы делятся на классы по среде обитания и на подклассы по способу заражения. По **среде обитания** существует четыре класса вирусов:

– скриптовые;

– загрузочные;

– файловые;

– макро-вирусы.

Рассмотрим их немного подробнее.

Скриптовые вирусы пишутся на скрипт-языках (*PHP*, *JS*, *VBS*, *BAT* и тому подобных). Они способны заражать файлы достаточно большого диапазона расширений: от .exe до *.html*.

Загрузочные вирусы обитают в загрузочных секторах дисков, дискет и других съемных цифровых носителей информации, а также в загрузочных секторах или в главной загрузочной записи винчестера. Их принцип действия основан на том, что при запуске ОС после включения или перезагрузки компьютера, после окончания процесса проверки оборудования (памяти, процессора, дисков, шин и так далее) *BIOS* компьютера считывает первый физический сектор загрузочного диска (*A:, C:* или *CD-ROM* в зависимости от параметров, установленных в *BIOS Setup*) и передает на него управление. Если загрузочный сектор дисков заражен, то при загрузке системы первым получает управление вирус. Таким образом, вредоносное ПО «заставляет» ОС при ее запуске считать из памяти код вируса и передать управление ему.

Попадание вирусов на жесткий диск компьютера может происходить следующими путями:

– вместо кода главной загрузочной записи винчестера (*MBR)* может быть записан вирус;

– в *boot*-сектор загрузочного диска вместо оригинального кода может быть записан код вируса;

– в главной загрузочной записи винчестера может быть изменен адрес активного сектора в таблице разделов диска.

Почти во всех случаях загрузочный вирус переносит оригинальный boot-сектор или MBR в другой свободный сектор диска. Если длина вируса больше длины *boot*-сектора, то в заражаемый сектор записывается только первая часть вируса, а остальное тело размещается в других свободных секторах диска. При загрузке ОС первым управление получает вирус, выполняет свою функцию и затем передает управление загрузочной записи.

Файловые вирусы – это класс вирусов, которые в качестве среды обитания используют исполняемые файлы компьютера. Они заражают исполняемые файлы или создают файлы-двойники. По способу заражения файловые вирусы экспертами лаборатории Касперского делятся на три подкласса.

Перезаписывающие вирусы. Вирусы данного подкласса (***Overwriting***) заменяют оригинальный код исполняемого файла программы на свой. После этого программа, естественно, перестает работать, так как ее исполняемого файла больше не существует. По этой же причине такие файлы после заражения восстановлению не подлежат и требуют полной переустановки. Обнаружить такой вирус можно и без использования антивирусных средств, так как неработоспособность систем или программ в результате работы перезаписывающего вируса нетрудно заметить.

Паразитические вирусы. К подклассу паразитических вирусов (***Parasitic***) относятся вирусы, которые дописывают свой код в файл-жертву, оставляя его работоспособным. В зависимости от того, в какое именно место файла-жертвы паразитирующий вирус дописывает свой код, данные вирусы делятся на вирусы, которые записываются в начало (*prepending*), середину (*inserting*) или конец (*appending*) файлов. Если тело вируса записывается в начало файла, то вирус смещает начало оригинального кода файла, дописывая перед ним свой код. Но он может сделать и иначе – перенести оригинальный код начала заражаемого файла в конец, а собственный код записать на освободившееся место. Во всех этих случаях после запуска файла первым управление получает вирус, а после исполнения своей деструктивной функции вирус передает управление оригинальному файлу. При внед-

рении паразитирующих вирусов в середину файла-жертвы также может происходить перенос части оригинального кода файла из середины в его конец, но возможно и копирование кода вируса в заведомо неиспользуемые данные файла-жертвы (*cavity*-вирусы). При внедрении в конец файла-жертвы код вируса просто дописывает себя в конец файла. Но нужно понимать, что во всех случаях головная часть файла-жертвы обязательно изменяется таким образом, что при запуске файла пользователем первым управление получает вирус, а потом уже зараженный файл. Чаще всего переносом паразитирующим вирусом точки входа на участок, который принадлежит ему, является дописывание кода файла-жертвы к своему коду. Но существует и другой способ, при котором в начало файла-жертвы вирусом дописывается команда передачи управления ему. В этом случае файл-жертва стартует с оригинальной точки входа, а затем, по этой добавленной команде, управление передается коду вируса.

Вирусы-компаньоны. Этот подкласс вирусов своим названием (*Companion*) обязан тому, что при внедрении в файловую систему компьютера они создают копию файла-жертвы с тем же именем, но иным расширением. Код файла-жертвы при этом не меняется, а изменяется только расширение его имени, например, с *.exe* на *.com*. А файл вируса получает имя и расширение в точности как у файла-жертвы. Пользователь запускает нужную ему программу, но реально он запускает на исполнение вирус, который выполнят свою деструктивную функцию, заражая еще несколько исполняемых файлов, а потом передает управление программе, которую хотел запустить пользователь. Если пользователь обратит внимание на время запуска программы, то он сможет обнаружить вирус-компаньон даже без использования антивирусных средств, так как время запуска будет увеличено в несколько раз.

Существуют и другие подклассы классических вирусов по способу заражения, которые встречаются очень редко, поэтому просто перечислим их:

- вирусы, заражающие объектные модули (*OBJ*);
- вирусы, заражающие библиотеки компиляторов (*LIB*);
- вирусы, заражающие исходные тексты программ.

Макровирусы обычно заражают документы *MS Office*, добавляя свой код в область макросов документа. Расположение кода вируса в документах разных приложений *MS Office* может быть разным.

1.1.3. ТРОЯНСКИЕ ПРОГРАММЫ

К ВПО данного вида (троянам, троянцам), по классификации лаборатории Касперского, относятся вредоносные программы, которые придают программному обеспечению информационных систем недекларированные возможности, т.е. новые свойства, не санкционированные пользователем. Этими свойствами могут быть уничтожение, кража или искажение информации и другие, им подобные. По этой причине трояны являются самым опасным из вредоносного ПО. Кроме того, они же являются и самым многочисленным видом ВПО в компьютерной среде. Причиной этого является то, что существует большое количество конструкторов троянских программ, которые позволяют даже пользователям, далеким от программирования, создавать собственное троянское ПО. Рассмотрим некоторые классы троянских программ подробнее.

Троянские утилиты удаленного администрирования. Утилиты удаленного администрирования присутствуют на любом компьютере. Троянские программы этого класса почти ничем от них не отличаются, только они находятся и действуют в системе без ведома законного пользователя. При загрузке и инсталляции эти трояны не выдают никаких уведомлений, но после успешного внедрения на компьютер-жертву их обладатель получает возможность удаленно администрировать его операционную систему. Поэтому программы такого класса являются самыми опасным из всего троянского ПО. Некоторые троянские утилиты удаленного администрирования могут распространяться по сети аналогично сетевым червям, но только не самостоятельно, а в результате выполнения соответствующей команды своего обладателя.

Похитители паролей. Само название говорит о том, что троянские программы данного класса позволяют своему обладателю похищать с компьютера-жертвы пароли. Но на самом деле они способны похищать с зараженного компьютера и другую подобную информа-

цию. После инсталляции на компьютер-жертву, такая программа (*Trojan-PSW*) сразу приступает к поиску файлов, которые содержат информацию нужную ее обладателю. Это могут быть файлы определенных типов: номера счетов, коды активации другого ПО, ключи шифрования или электронной подписи и тому подобных.

Интернет-кликеры. Предназначением троянских программ этого класса (*Trojan-clicker*) является обращение без ведома законного пользователя зараженного компьютера к различным удаленным ресурсам. С этой целью троянец может посылать соответствующие команды браузерам или подменять адреса нужных сайтов с целью увеличения их посещаемости, увеличения количества просмотров публикаций, организации DoS-атак или привлечения потенциальных жертв для заражения вредоносным ПО. Компьютер, который по команде обладателя такой троянской программы без ведома пользователя начинает обращаться к интернет-ресурсам, получил название компьтер-зомби.

Загрузчики. Троянские программы этого класса (*Trojan-Downloader*) выполняют загрузку ПО на компьютер-жертву без ведома его пользователя. Естественно, загружаемое ПО является вредоносным. Загруженное на компьютер-жертву программное обеспечение может сразу инсталлироваться или записываться в автозагрузку.

Установщики. Предназначением троянских программ этого класса (*Trojan-Dropper*) является инсталляция вредоносного программного обеспечения на компьютер-жертву. ВПО этого класса всегда состоит из основного кода и файлов. Основной код является троянцем, а файлы содержат ВПО, которое он должен установить. Установщик сначала записывает их в каталог временных файлов, а затем инсталлирует. Установка может производиться как незаметно для пользователя, так и с выдачей сообщения об ошибке.

Троянские прокси-серверы. Этот класс троянцев (*Trojan-Proxy*) обеспечивает своему обладателю скрытный доступ к разным хостам, как правило, для рассылки спама.

Шпионские программы. Троянцы этого класса (*Trojan-Spy*) могут наблюдать за ничего не подозревающим пользователем, записывая информацию, которую он набирает на клавиатуре, делая скрины экрана, передавая изображение и звук с веб-камеры и так далее. Есть

в этом классе троянов и «многоцелевые» программы, которые одновременно могут выполнять несколько функций, например, шпионить за пользователем и предоставлять *proxy*-сервис удаленному злоумышленнику.

Скрытие присутствия в операционной системе. Предназначением троянцев этого класса (*Rootkit*) является сокрытие присутствия в системе зараженного компьютера некоторых объектов, тип которых задается заранее. Это могут быть определенные файлы, ключи, процессы и так далее. В соответствии с классификацией «Лаборатории Касперского», ВПО типа *Rootkit* обладает самым младшим поведением среди вредоносных программ. Это означает, что если *Rootkit* имеет троянскую составляющую, то она будет распознана как *Trojan*.

Архивные бомбы. ВПО этого класса (*ArcBomb*) отнесено к троянскому лишь условно. Это, по своей сути, обычный архив, который при обработке вызывает «нестандартную» реакцию архиватора. В результате этого жесткий диск может заполниться большим количеством ненужной информации и компьютер может прекратить или существенно замедлить свою работу.

Любой архиватор позволяет упаковать файл значительных размеров в архив небольшого размера, если изначально этот файл содержал повторяющиеся данные. А при разархивации этот архив снова выдаст файл большого размера. На этом простом свойстве архиваторов и основан принцип действия архивных бомб.

Различают три типа *ArcBomb*:

- повторяющиеся данные;
- некорректный заголовок архива;
- одинаковые файлы в архиве.

Оповещение об атаке, увенчавшейся успехом. Трояны этого класса (*Trojan-Notifier*) обычно входят в состав многокомпонентных наборов ВПО**.** Они предназначены для оповещения своего обладателя об успешной установке троянских компонентов в атакуемую систему путем отправления ему электронного письма или специального обращения к его веб-странице, содержащего информацию о зараженном компьютере: *IP*-адрес, номер открытого порта, адрес электронной почты и тому подобной.

1.1.4. ПРОЧИЕ ВРЕДОНОСНЫЕ ПРОГРАММЫ

Сетевые черви, классические вирусы и троянские программы не являются исчерпывающими видами вредоносного ПО. Для завершения его классификации лабораторией Касперского введен еще один вид – прочее ВПО. К этому виду относятся различные программы, которые не предназначены для непосредственной реализации каких-либо угроз информационным системам. Они предназначены, например, для поиска уязвимостей в системе защиты компьютеров, создания другого вредоносного ПО, организации *DoS*-атак и тому подобного.

Программы для организации сетевых атак. Данный класс прочего ВПО предназначен для организации удаленных атак, направленных на отказ в обслуживании (*DoS, DDoS*). При такой атаке на адрес жертвы отправляется большое количество запросов или пакетов, с которыми оборудование не справляется и дает сбой или, как говорят, «зависает». В данном классе различают два вида программ. ВПО первого вида осуществляет атаки непосредственно с компьютера злоумышленника по его приказу. Это, так называемая, *DoS*-атака. ВПО второго вида организует распределенную атаку с использованием зараженных компьютеров-зомби. При этом пользователь зараженного хоста обычно не подозревает о том, что его компьютер является участником *DDoS*-атаки.

Взломщики удаленных компьютеров. Обычно ВПО этого класса (*Exploit, Hacktool*) предназначено для поиска слабых мест в системе защиты компьютеров, подключенных к сети, для реализации различного рода атак на них.

«Замусоривание» сети – засорение каналов компьютерной сети бесполезной информацией (*Flood*). Иногда это делает не ВПО, а люди, рассылая друг другу в огромном количестве различные «письма счастья».

Конструкторы. ВПО этого класса, как следует из его названия (***Constructor***), используется для создания другого ВПО, как правило, троянского, людьми, которые не являются программистами.

Фатальные сетевые атаки. К этому классу ВПО относятся вспомогательные компьютерные программы (утилиты) в составе общего программного обеспечения ЭВМ, которые отправляют специ-

альным образом оформленные запросы на атакуемые компьютеры в сети, что приводит к прекращению работы атакуемой системы. В данных атаках используются уязвимости в системном и прикладном ПО, что позволяет сетевому запросу специального вида вызвать критическую ошибку в атакуемом приложении или в системе.

Введение пользователя в заблуждение. ВПО данного класса (*Bad-Joke*, *Hoax*) даже трудно назвать вредоносным, так как оно не наносит абсолютно никакого вреде ни программам, ни данным. Оно, скорее, имеет целью запугивание пользователя, выдавая ему различные сообщения, типа «Приступаю к форматированию винчестера».

Шифровальщики вредоносного ПО. К этому классу прочего ВПО (*FileCryptor, PolyCryptor*) относятся хакерские утилиты, которые скрывают иное вредоносное ПО от антивирусных средств.

«Полиморфы». Этот класс прочего ВПО (*PolyEngine*), также как и ВПО класса *Bad-Joke* или *Hoax,* трудно назвать вредоносным. Их код не содержит команды, направленные на порчу программ, данных, самокопирование и тому подобное. Их задача затруднить антивирусным средствам обнаружение вредоносного ПО.

Таким образом, не надо думать, что вредоносная программа – это обязательно опасная вещь, среди них нередко можно встретить вполне безвредные экземпляры.

1.2. МЕТОДЫ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.2.1. КЛАССИФИКАЦИЯ АНТИВИРУСНЫХ ТЕХНОЛОГИЙ

Антивирусные технологии включают в себя методы обнаружения вредоносного ПО и методы его нейтрализации.

Существует несколько основополагающих методов обнаружения вредоносных программ:

- сканирование;
- обнаружение странного поведения программ;
- обнаружение при помощи эмуляции;
- эвристический анализ:
- обнаружение изменений;
- резидентные мониторы.

Основополагающими методами нейтрализации ВПО являются:

- метод «белого списка»;
- вакцинирование программ;
- аппаратная защита;
- защита, встроенная в *BIOS* компьютера.

Антивирусных средств, которые реализовывали бы все перечисленные методы обнаружения и нейтрализации ВПО, до настоящего времени не существует. Имеющиеся в нашем распоряжении антивирусы могут реализовывать лишь некоторые комбинации этих методик и почти все они обеспечивают восстановление зараженных файлов и загрузочных секторов в случае, если это вообще возможно.

Развитие антивирусных технологий происходит параллельно с совершенствованием вредоносного ПО и к настоящему времени они уже стали достаточно сложными и изощренными.

Единой общепринятой классификации антивирусных технологий, как и классификации вредоносных программ, пока не существует. Но можно считать фактическим стандартом такой классификации деление антивирусных технологий на реактивную защиту и проактивную защиту по типу нейтрализуемой угрозы – соответственно, известной или неизвестной.

Реактивная защита основана на знании уникальных особенностей фрагментов кода вредоносных программ.

Проактивная защита базируется на знании характерных для вредоносных программ неуникальных особенностей кода и поведения.

Довольно часто компании, занимающиеся защитой ЭВМ от вредоносного ПО, используют для классификации антивирусных технологий стандартный подход – классификационный признак и основные типы антивирусных технологий по этому признаку. Воспользуемся тем же подходом.

По признаку анализируемых свойств угроз и предположительно зараженных объектов различают следующие способы обнаружения ВПО:

- анализ кода подозрительных объектов;
- анализ поведения подозрительных объектов;
- отслеживание изменений объектов защищаемой ЭВМ.

По режиму осуществления защиты:

– постоянный мониторинг за процессами на компьютере и в сети с обнаружением угроз в момент открытия зараженного файла;

– сканирование компьютера по запросу пользователя;

– сканирование компьютера по расписанию.

Существует также классификационное деление антивирусных технологий по методу «белых» и «черных» списков.

Первый из этих методов предполагает разрешение активности лишь проверенных программ, содержащихся в списках, разрешенных к запуску. Активность всех остальных объектов блокируется.

Второй метод предполагает блокирование или ограничение активности только лишь объектов, которые содержатся в базах угроз, и разрешение активности остальных объектов.

Существует и подход, который использует «черные» списки для обнаружения угроз и «белые» списки для коррекции результатов детектирования и минимизации ложных срабатываний.

Знание современных технологий обнаружения и нейтрализации ВПО позволяет правильно выбирать средства защиты от ВПО. Полноценное антивирусное средство должно и обеспечивать обнаружение и нейтрализацию любых угроз. А для этого оно должно реализовывать весь комплекс необходимых для этого технологий, которые мы рассмотрим подробнее.

1.2.2. СКАНИРОВАНИЕ

Для реализации данной технологии разработчики должны составить и периодически пополнять базу сигнатур вредоносных программ. При сканировании компьютера антивирусное ПО формирует сигнатуру каждого проверяемого объекта и сравнивает его с базой сигнатур. При обнаружении совпадений сигнатур антивирус автоматически или с запросом выполняет одно из следующих действий в отношении подозрительного объекта:

– блокирует;

– удаляет;

– помещает в специальное хранилище (карантин), которое не позволит вредоносному коду распространяться и выполнять свои деструктивные функции;

– лечит, удалив из него вредоносный код.

Обычно поставщики ПО производят регулярное обновление антивирусной базы. Для успешного своевременного обновления базы желательно включить автоматическое обновление.

На современном рынке антивирусных продуктов представлены средства защиты от ВПО, использующиея разные технологии и конструктивные решения. Некоторые средства защиты от ВПО могут использовать сразу несколько ядер в целях обеспечения более результативного поиска вредоносных программ.

Многие современные средства защиты от ВПО реализуют технологию проверки файла непосредственно в момент, когда пользователь обращается к нему. Почти во всех продуктах пользователь может устанавливать периодичность проверки компьютера.

Создатели вредоносного ПО постоянно совершенствуют свои разработки, создавая так называемые «олигоморфические, полиморфические и метаморфические» вирусы с целью обхода антивирусных средств, используя их слабые места. Поэтому и средства защиты ЭВМ от ВПО должны постоянно совершенствоваться.

1.2.3. ОБНАРУЖЕНИЕ СТРАННОГО ПОВЕДЕНИЯ ПРОГРАММ

Принцип работы средств защиты от ВПО, которые реализуют этот метод, основан на выявлении подозрительных действий, совершаемых на компьютере другими программами. Эти средства способны обнаруживать вредоносные программы, сигнатуры которых не содержатся в их базе данных. При обнаружении подозрительных действий проверяемой программы антивирусное ПО их заблокирует и/или предупредит пользователя об этом событии.

Обнаружение странного поведения программ, как метод детектирования ВПО, в настоящее время получает все более широкое распространение и обычно реализуется в виде отдельных модулей в составе антивирусного ПО.

Несмотря на то, что этот метод показывает высокую эффективность, использующий его антивирус не всегда способен отличить вредоносный код от обычной уникальной активности. Такие ошибки приводят к блокировке нужных программ или попросту раздражать пользователя частыми предупреждениями об опасности, которой, возможно, и не существует.

1.2.4. ОБНАРУЖЕНИЕ ПРИ ПОМОЩИ ЭМУЛЯЦИИ

Принцип работы антивирусного ПО, реализующего этот метод обнаружения вредоносных программ, основан на том, что оно само имитирует выполнение кода некоторой программы до того, как эта программа реально будет запущена пользователем. И если исполнение программы после запуска пойдет не по тому алгоритму, который был эмулирован, то она будет распознана как вредоносная.

Данный способ демонстрирует высокую долю ложных срабатываний. По этой причине он не получил широкого распространения.

1.2.5. ОБНАРУЖЕНИЕ ИЗМЕНЕНИЙ

Программные антивирусные средства, реализующие метод обнаружения изменений, получили название программы-ревизоры. Принцип их работы основан на том, что любая вредоносная программа после попадания на компьютер неизбежно приводит к изменениям на его винчестере. Это может быть дописывание вирусом своего кода в исполняемый файл, добавление вредоносной программы в автозагрузку, изменение загрузочного сектора, создание файла-двойника и тому подобное.

Достаточно предварительно запомнить характеристики тех областей диска, которые могут быть подвержены изменениям при заражении, а потом периодически сверять их текущее состояние с запомненным. В случае обнаружения изменений программа-ревизор сообщит об этом пользователю.

Обычно такие программы запоминают содержание главной загрузочной записи, образы загрузочных секторов логических дисков, контрольные суммы или сигнатуры всех контролируемых файлов, информацию о структуре каталогов и номера поврежденных кластеров

диска. Иногда запоминаются также объем оперативной памяти, количество подключенных к компьютеру дисков и их параметры.

Программы-ревизоры могут успешно детектировать значительное количество вредоносного ПО, даже ранее неизвестного. Но заражение файлов программ, которое происходит лишь при их копировании, программы-ревизоры обычно не обнаруживают, так как не располагают информацией о параметрах файлов до копирования.

Однако нужно иметь в виду, что не все изменения на диске компьютера вызваны вредоносными программами. Загрузочная запись может измениться при обновлении версии ОС, а файл автозагрузки может измениться при установке нового ПО. Программы-ревизоры не смогут помочь также при записи пользователем нового файла, который уже содержит вирус. Однако заражение вирусом других программ, запомненных ревизором, будет успешно детектировано.

1.2.6. ЭВРИСТИЧЕСКИЙ АНАЛИЗ

Эвристический анализ позволяет обнаруживать ранее неизвестные вредоносные программы. В этом он схож с методом обнаружения изменений. Но эвристический анализ не требует предварительного сбора данных о файловой системе.

Антивирусные средства, реализующие данный метод, проверяют программы и загрузочные сектора дисков с целью обнаружения характерных для вредоносного ПО уникальных участков кода.

Но и этот метод пока имеет высокий процент ложных срабатываний. Поэтому, когда эвристический анализатор выдаст сообщение об обнаружении вредоносного кода в файле или загрузочном секторе, не следует этому безоговорочно доверять, а проверить такие «находки» при помощи последних версий антивирусного ПО или направить их для исследования экспертам компании, которая занимается защитой ЭВМ от вредоносных программ.

1.2.7. РЕЗИДЕНТНЫЕ МОНИТОРЫ

Резидентными мониторами называют класс антивирусных программ, которые постоянно находятся в оперативной памяти компьютера и отслеживают все происходящие изменения. Например, если

какая-то программа попытается изменить загрузочный сектор жесткого диска, исполняемый файл или оставить в оперативной памяти резидентный модуль, то резидентный монитор сразу это обнаружит и сообщит об этом пользователю.

Большинство резидентных мониторов выполняют также функции сканера, т.е. проверяют все запускаемые на исполнение программы на предмет содержания известных сигнатур ВПО. Такая проверка безусловно замедляет загрузку программы, но является эффективным средством обнаружения известных экземпляров ВПО.

Но резидентные мониторы не нашли широкого применения из-за большого количества имеющихся у них недостатков. Во-первых, они демонстрируют большую долю ложных срабатываний. Так обычная команда *LABEL* изменяет данные в загрузочном секторе, что приводит к срабатыванию монитора. В результате пользователю необходимо каждый раз решать, вызвано ли данное срабатывание наличием вируса или нет. На практике это приводит к тому, что пользователь отключает резидентный монитор. Во-вторых, резидентный монитор постоянно находится в оперативной памяти компьютера и занимает в ней место, которое могло бы быть доступно другим программам.

1.3. МЕТОДЫ НЕЙТРАЛИЗАЦИИ ВРЕДОНОСНЫХ ПРОГРАММ

1.3.1. МЕТОДЫ «БЕЛОГО СПИСКА» И «ЧЕРНОГО СПИСКА»

Суть метода «белого списка» заключается в том, что, реализующее его защитное ПО осуществляет проверку всех программ и выполняемых ими действий, при необходимости пресекая их, за исключением тех, что включены администратором в список доверенных. К достоинствам этого метода можно причислить высокую оперативность работы, низкую ресурсоемкость, высокую достоверность. Данный метод позволяет сделать так, чтобы на защищаемом компьютере могли выполняться только программы, находящиеся в списке разрешенных и блокировалась активность любого другого ПО, в том числе и вредоносного, в случае его успешного попадания на защищаемый компьютер. Некоторые разработчики присваивают

своим приложениям статус «надежное». Это значит, что они гарантируют его работоспособность и защищенность от ВПО, составляя таким образом свои «белые списки».

Но, тем не менее, самыми популярными на рынке средствами защиты от ВПО являются все же программы, реализующие принцип работы противоположного «черного списка». Главной причиной этого является то, что такой метод обязывает клиента оформлять подписку на услуги компании-разработчика поддерживать актуальность «черных списков», т.е. метод «черного списка» является прибыльнее, а значит и популярнее «белого списка».

1.3.2. ВАКЦИНИРОВАНИЕ ПРОГРАММ

Этот способ защиты программного обеспечения от вирусов основан на присоединении к защищаемой программе специального модуля контроля, следящего за ее целостностью. При заражении вакцинированной программы вирусом, модуль контроля обнаружит изменение ее контрольной суммы, размера исполняемого файла или другой характеристики и сообщит об этом пользователю.

В большинстве случаев вакцинирование не позволяет защитить программу от заражения. Некоторые вредоносные программы, например стеллс-вирусы, могут обмануть вакцину, так зараженные ими программы продолжают работу в обычном режиме, а вакцина не обнаруживает заражения.

1.3.3. АППАРАТНАЯ ЗАЩИТА

В настоящий момент программно-аппаратные средства, включающие в себя контроллер, подключаемый к одному из разъемов расширения компьютера, и специальное ПО, которое управляет работой этого контроллера, являются одним из самых надежных способов защиты ЭВМ от вредоносного ПО.

Контроллер, подключенный к системной шине компьютера, получает полный контроль над всеми обращениями к его дискам, а специальное ПО позволяет определить объекты, изменение которых невозможно. Это могут быть главная загрузочная запись, загрузочные сектора, файлы конфигурации, исполняемые файлы и другие объекты, определяемые пользователем.

1.1. Комплексы защиты от вредоносных программ

Наименование комплекса	Изготовитель
Virustrap	JAS Technologies of the Americas
C:Cure	Leprechaum Software International
V-Card	Digital Enterprises
Thunderbyte	Glynn International
Immune	Swabian Electronics Reutlingen
ExVira	Bugovics & Partner

Аппаратный уровень защиты компьютера не позволяет вредоносным программам замаскироваться. Как только вредоносная программа себя проявит, она сразу же будет обнаружена и заблокирована.

Аппаратно-программные комплексы защиты ЭВМ от вредоносных программ не только обнаруживают, но и пресекают активность любого ВПО. Кроме того, они защищают информационные системы от ошибок пользователя или действий злоумышленника, не позволяя удалить важную информацию, отформатировать диск, производить несанкционированное копирование информации и так далее.

Примером аппаратно-программного комплекса защиты ЭВМ от ВПО является отечественный комплекс *Sheriff*.

За рубежом аппаратно-программных средств защиты от ВПО выпускается много, но их цена значительно превышает цену *Sheriff*. В таблице 1.1 приведены некоторые примеры таких комплексов.

Практически все аппаратно-программные комплексы защиты компьютера от ВПО обеспечивают также защиту ресурсов компьютера от несанкционированного доступа, а также могут обеспечивать и другой дополнительный сервис.

1.3.4. ЗАЩИТА, ВСТРОЕННАЯ В *BIOS* КОМПЬЮТЕРА

Можно без преувеличений считать, что все современные системные платы компьютеров имеют в своем составе средства контроля обращений к главной загрузочной записи винчестера и к загрузочным секторам дисков, а это, по сути, является средством антивирусной

защиты на аппаратном уровне. Если какая-то программа попытается модифицировать содержимое загрузочных секторов диска, то это сразу будет обнаружено и предупрежденный пользователь сможет запретить либо разрешить эту процедуру.

Только не надо думать, что это является полноценной аппаратной защитой от ВПО, заменяющей специальные программно-аппаратные комплексы. Надо иметь в виду, что сам программный модуль, осуществляющий контроль загрузочных секторов, находится в памяти *BIOS,* и поэтому вредоносные программы имеют возможность его обойти, обращаясь непосредственно к портам их контроллера. Кроме того, некоторые вредоносные программы способны отключить антивирусный контроль, встроенный в *BIOS,* изменив содержимое некоторых ячеек в *CMOS*-памяти.

Таким образом, защита от ВПО, встроенная в *BIOS* компьютера, хотя и представляет собой защиту на аппаратном уровне, но не заменяет программно-аппаратные комплексы защиты от ВПО.

1.3.5. МЕТОДЫ УДАЛЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

Для полноценной защиты ЭВМ от вредоносных программ недостаточно ограничиться их обнаружением. С обнаруженным ВПО нужно что-то делать. Обычно, антивирусные средства, которые детектируют вредоносное программное обеспечение, способны его удалить. Для этого существуют, по крайней мере, два способа.

Первый способ применяется, когда обнаружен заведомо известный вредоносный код, который уже изучен и разработан алгоритм лечения зараженных им файлов, реализованный в антивирусной программе.

Второй способ необходим для лечения объектов информационной системы, зараженных ранее неизвестными вирусами. Для этого антивирус, до появления в компьютере вирусов, должен провести анализ всех исполняемых файлов и загрузочных секторов и сохранить о них много различной информации.

Файлы и загрузочные секторы дисков, зараженные перезаписывающим вирусом, не могут быть вылечены с использование первой методики, но нередко могут быть восстановлены при помощи второй.

Если же зараженные файлы восстановить антивирусной программой невозможно, то можно попытаться восстановить их с дистрибутива или резервной. Наконец, их можно удалить и заново инсталлировать.

Если обнаружено заражение вредоносной программой исполняемых файлов с расширением *.com* или *.exe*, то нужно обязательно проверить и все другие типы исполняемых файлов с расширениями *.zip*, *.pak*, *.pif*, *.doc*, *.lzh*, *.dll*, *.pgm*, *.sys*, *.ovi*, *.ovl*, *.ovr*, *.bat*, *.bin*, *.drv*, *.lib*, *.bak*, *.arj*, *.zip*. Лучше проверить все файлы на жестких дисках, так как вирус может изменить расширение имени файла. Например, файл *game.exe* может быть переименован в *game.ex_*. Такой файл проверяться не будет. Но если через некоторое время его переименовать обратно, то вирус снова сможет продолжить выполнение своих деструктивных функций в компьютере.

Антивирусная программа при последующих запусках повторно собирает данные об исполняемых файлах и сопоставляет их с данными, полученными ранее. При обнаружении несовпадения делается вывод о возможном заражении этого файла, и антивирусная программа попытается его восстановить, используя информацию об этом файле, полученную ранее.

Восстановить главную загрузочную запись и загрузочные сектора после обнаружения их заражения значительно труднее. Если антивирусная программа не может их восстановить, то придется это делать пользователю с помощью команд *fdisk, format, sys.*

Некоторые вредоносные программы после заражения компьютера становятся частью его операционной системы. В случае после простого удаления зараженного файла и последующей его переустановки, ОС может оказаться неработоспособной. Здесь может помочь первая методика.

Примерами таких вирусов могут служить вирусы типа *VolGU*, а также загрузочные вирусы *OneHalf*.

Так, вирус *OneHalf* при загрузке компьютера шифрует данные, находящиеся на жестком диске. И пока такой вирус является резидентным, он может перехватывать все обращения к жесткому диску. Если какая-либо программы либо процесс попытается считать информацию с уже зашифрованного сектора, то вирус расшифрует ее. А после удалении вируса *OneHalf* информация на зашифрованной части жесткого диска станет недоступной.

Вирусы типа *VolGU* не шифруют данные, но они не менее опасны. В секторах винчестера хранятся не только данные, записанные в них, но и контрольные суммы всех байт сектора, используемые для проверки целостности данных.

При обращении программы или процесса к винчестеру, считываются и записываются только данные, а контрольная сумма корректируется автоматически. Вирус *VolGU* при записи данных на диск портит контрольные суммы секторов. Пока данный вирус активен, он позволяет считывать секторы с неправильной контрольной суммой, а после его удаления секторы с испорченной контрольной суммой не будут читаться. ОС сообщит об ошибке чтения с жесткого диска («сектор не найден»).

Контрольные вопросы

1. По какому принципу вредоносное ПО подразделяется на виды и классы?
2. Поясните разделение антивирусной защиты на реактивную и проактивную.
3. Поясните суть защиты от ВПО по методам «белого» и «черного» списков.
4. Как можно обнаружить вредоносное ПО путем анализа странного поведения программ?
5. Поясните суть аппаратно-программной защиты ЭВМ от ВПО, ее достоинства и недостатки.
6. Поясните особенности удаления с компьютера вредоносных программ.

2. ТЕХНОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1. ПОКАЗАТЕЛИ ТЕХНОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ

При проектировании и реализации сложных программных средств и баз данных практически невозможно полностью избежать ошибок. В связи с этим важно установить метрику и определить ее значения, которые объективно отражали бы уровень безопасности информационных систем в реальной или наиболее вероятной совокупности потенциальных дефектов.

Понятие «технологическая безопасность информационных систем» похоже на понятие «надежность программно-аппаратных комплексов». Поэтому для количественной оценки технологической безопасности можно использовать уже известные метрики теории надежности. Но в теории надежности учитываются все отказы, а в показателях технологической безопасности учитываются только те отказы, которые влияют на нарушение конфиденциальности, целостности и доступности данных. Таких отказов бывает значительно меньше. Известные значения показателей надежности программных продуктов и баз данных могут выступать ориентирами при оценке безопасности критических ИС. Рассмотренные ниже способы оценки технологической безопасности информационных систем основываются на концепции баз данных и надежности программ.

Безопасность информационной системы можно охарактеризовать величиной ущерба, который может быть причинен дестабилизирующими факторами и реализацией угроз безопасности. Однако описать и оценить вероятный ущерб информационных систем при нарушении безопасности критических ИС разных классов весьма проблематично. Поэтому угрозы целесообразно характеризовать интервалами времени между их проявлениями. Этот показатель аналогичен наработке на отказ в теории надежности.

Нарушения работоспособности программного обеспечения при условии безотказности аппаратуры в основном происходят из-за противоречий между сочетанием исходных данных, подлежащих обработ-

ке, и программой, которая эту обработку реализует. Очевидно, что если исходные данные будут теми же, что были при испытаниях и отладке ПО, то работоспособность ИС можно гарантировать. Но реальные исходные данные могут существенно отличаться как от заданных техническим заданием, так и от тех, которые применялись при тестировании и отладке ПО. Как результат, вероятны различные аномалии в функционировании ПО, которые могут завершаться сбоями и отказами.

Давайте определим понятия «сбой» и «отказ» программного обеспечения. Еще сравнительно недавно аномалия в работе ПО, которая устранялась перезапуском программы называлась сбоем, а если требовалась переустановка программы, то это называлось отказом. Но с появлением предоставления софта как услуги (*SaaS*), основным классификационным признаком, разделяющим сбои и отказы программного обеспечения, стал не способ, а время восстановления работоспособности: если работоспособность восстанавливается за время не превышающее заданное, то это сбой, а если для восстановления работоспособности требуется время, превышающее заданное, то это отказ.

Одни аномалии в работе ПО совершенно не отражаются на безопасности функционирования информационных систем, последствия же иных видов сбоев и отказов ПО могут квалифицироваться как нарушение безопасности функционирования информационной системы с критическими, а иногда и с катастрофическими последствиями. Универсальным и легко измеряемым показателем технологической безопасности ИС остается время восстановления работоспособности. В этом случае приближенно такие отказы в ИС можно выделять по превышению какого-либо возможного времени восстановления работоспособности, отличающегося от порогового времени, которое разделяет отказы и сбои.

В теории надежности состояние объекта является работоспособным, если этот объект способен выполнять заданные ему функции с параметрами, установленными требованиями технической документации. Надежность является внутренним параметром систем, проявляющимся лишь во времени. Критерии качества становятся преимущественно стохастическими и динамическими, характеризующими работу крупных групп программ или информационной системы в целом.

Измеряемые интегральные показатели качества программ в таком случае имеют более определенный характер и могут с достаточной точностью экспериментально оцениваться.

Устойчивость (или живучесть) наиболее всеобъемлюще характеризует способность информационной системы безотказно функционировать при появлении отказов и сбоев. Она зависит от степени неустраненных ошибок и способности информационной системы реагировать на проявления ошибок таким образом, чтобы это никак не отражалось на показателях безопасности и надежности. Показатель безопасности определяется эффективностью контроля доступа к данным, степенью обеспечения их целостности и конфиденциальности и селекцией подлинных данных, поступающих из внешней среды, средствами обнаружения отклонений и эффективностью процессов восстановления работы информационной системы.

Восстанавливаемость программы после сбоя или отказа характеризуется полнотой восстановления ее функционирования после перезапуска. На перезапуск программы требуются ресурсы ЭВМ и время, но он должен обеспечивать возобновление нормального функционирования информационной системы. Длительность и полнота восстановления функционирования программного обеспечения после сбоев и отказов отражают безопасность информационной системы и ее качество.

Обобщение частоты отказов и длительности восстановлений ПО производится в коэффициенте готовности. Этот показатель имеет смысл вероятности иметь восстанавливаемую систему в рабочем состоянии в выбранный произвольно момент времени. Коэффициент готовности определяется как часть времени полезной работы информационной системы на достаточном интервале наблюдения, содержащем восстановления и отказы.

$$K_{\text{гот}} = T_{\text{раб}} / T_{\text{набл}},$$

где $T_{\text{раб}}$ – суммарное время полезной работы; $T_{\text{набл}}$ – общее время наблюдения, включающее время полезной работы и время восстановления работоспособности ($T_{\text{набл}} = T_{\text{раб}} + T_{\text{восст}}$).

Применение основных понятий теории надежности для оценки надежности и безопасности сложных систем позволяет получить ряд

хорошо измеряемых, четких интегральных качественных показателей программ. Данные критерии применяются в основном при испытании информационной системы и на конечных этапах комплексной отладки. Их практически нельзя применять для оценки качества программных компонент, решающих частные функциональные задачи.

2.2. МЕТОДЫ ОПРЕДЕЛЕНИЯ ПОКАЗАТЕЛЕЙ ТЕХНОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

2.2.1. ОСНОВНЫЕ ТРЕБОВАНИЯ К СРЕДСТВАМ И ВИДЫ ТЕСТИРОВАНИЯ ДЛЯ ОПРЕДЕЛЕНИЯ ТЕХНОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

Существенно способствовать повышению технологической безопасности сложных, критических информационных систем позволяет систематизация типов тестирования и их упорядоченное проведение при испытаниях. Данные виды тестирования направлены на дифференцированное выявление дефектов определенных классов. Целесообразно для всех видов тестирования создавать методику его проведения с указанием ожидаемых результатов и контролируемых параметров. При заключительных испытаниях или сертификации, кроме того, должно проводиться обобщенное тестирование при наиболее широком варьировании тестов в условиях, соответствующих реальной эксплуатации.

Для выявления ошибок функционирования в нормальных условиях, установленных техническим заданием на стандартную версию программного средства, служит тестирование обстоятельности решения функциональных задач при стандартных исходных данных. Первичным эталоном здесь являются цели и задачи разработки ПО. В соответствии с этими целями и задачами разрабатывается исчерпывающее формализованное техническое задание и спецификация требований на совокупность программ, которые являются основными эталонами при создании тестов данного вида. В системах реального времени тесты включают в себя динамические и стохастические данные. Эти данные имитируются моделями реальных объектов внешней среды. Результа-

ты этого тестирования обрабатываются и сопоставляются с эталонами преимущественно автоматически. Доля тестов может содержать детерминированные исходные данные, для анализа которых достаточно часто применяются различные системы графического отображения.

Стрессовое тестирование, т.е. тестирование функционирования программ в критических ситуациях, по логике и условиям решения задач предназначено для испытаний исполнения программ в нерядовых ситуациях, которые нечасто возникают на практике, но очень важны для безопасной работы информационно-управляющих систем. Для создания таких тестов разрабатываются сценарии критических комбинаций значений исходных данных и условий решения задач, при которых появляется необходимость проверить работу программ и можно ожидать неточности результатов и отказы в работе. Такие нештатные, стрессовые сочетания подготавливаются вручную или их реализация предусматривается в составе данных стохастических тестов в реальном времени. Особенная важность проверки при критических ситуациях определяется возможностью проявления ошибок при работе программного средства в конкретных условиях. Следовательно, при тестировании часто применяются имитаторы внешней среды, которые автоматически подготавливают исходные данные, и средства контроля, реагирующие на неестественные результаты исполнения тестируемых программ, отражающиеся на безопасности программ.

Тестирование правильности применения ресурсов производительности и памяти вычислительной системы необходимо для определения оценки безопасности выполнения программ при перегрузках производительности и памяти. Тестирование происходит, в основном, в реальном времени в стохастическом режиме по подготовленным сценариям, которые создают перегрузки одного из множества ресурсов системы. Преимущественно полно проверяются буферные накопители выдачи и приема информации внешним пользователям и использование массивов в базе данных. Данной проверке подлежит изменение надежности, безопасности и качества работы информационной системы вследствие пропусков в обработке сообщений, увеличения длительности ожидания перед их растягиванием или обработкой периодов решения задач. Имитация проверок производится предпочтительно по сценариям автоматически, создающим критические условия для

конкретной проверки. В результате проверки устанавливаются реальные характеристики информационной системы на какой-либо вычислительной системе по вероятной интенсивности решения отдельных типов задач и обработке различных сообщений, а также по пропускной способности решения всех задач. При кратковременных перегрузках памяти или производительности должна выполняться защита от отказов при следующем снижении загрузки и восстановление обычного режима.

Проверка параллельного выполнения программ применяется для нахождения снижений безопасности и надежности, вызванной разноречивым применением промежуточных и исходных данных, а также, при параллельном исполнении программ, устройств вычислительной системы. Проверка производится обычно стохастически с главной задачей осуществить проверку при разных комбинациях исполнения фрагментов программы одновременно работающими процессорами. Важной задачей является обнаружение всех тупиковых ситуаций. Небольшая вероятность возникновения данных ситуаций может вести к необходимости генерации большого количества стохастических тестов. Проверка параллельно выполняемых программ, как правило, требует большого числа исходных данных, содержащих случайные и детерминированные составляющие. Такие данные подготавливаются автоматически по сценариям более критических сочетаний данных.

Проверка эффективности защиты от модификаций исходных данных необходима для поиска ошибок в программах, проявляющихся при неверных или искаженных данных. Проверка происходит при небольших искажениях данных, соответствующих нормированному увеличению числа в них ошибок, а также при полном случайном искажении данных. Разумно измерять вероятности появления искажений и другие их характеристики. При проверке возникают ситуации нарушения работоспособности ИС и понижение безопасности ее работы в зависимости от интенсивности искажений. Искажения данных происходят обычно стохастически, однако, иногда могут быть необходимы коррелированные и детерминированные искажения данных.

Проверка для оценки эффективности защиты от сбоев аппаратуры и необнаруженных ошибок данных и программ служит для оперативного восстановления (рестарта) при различных неумышленных иска-

жениях работы программ и определения качества средств программного контроля. Стохастическая проверка средств рестарта происходит в процессе расчета показателей надежности функционирования программных продуктов. При этом оценивается средняя длительность восстановления и вероятность обнаружения отказов. Специальный тест оценки эффективности защиты позволяет определить возможность выявления каждого типа сбоев и соответствующую этому сбою длительность восстановления нормального функционирования. Для этого разрабатываются планы имитации аппаратных сбоев, искажений исходных данных и программных ошибок, которые вызывают включение средств оперативного восстановления и программного контроля. Исходя из этого обнаруживаются ошибки программ восстановления или контроля, а также определяются динамические параметры. Сложность такого вида тестирования определяется сложностью регламентированного ввода сбоев вычислительных систем и трудностью имитации возникновения ошибок в разработанных программных продуктах. Помимо этого, для получения статистических оценок нужно искусственное повышение интенсивности возникновения отказовых ситуаций.

Проверка для измерения значений надежности базовых версий информационных систем предназначено для установления безопасности при реальном функционировании программного обеспечения и базовых показателей надежности. При проверке в критических и типовых условиях устанавливаются значения коэффициента готовности, наработки на отказ, длительности восстановления и других показателей. Для сложных систем реального времени формируются многочасовые прогоны при стохастических исходных данных, при которых выделяются нарушения работоспособности программ и регистрируются искажения результатов. При такой проверке особое значение имеет соотношение критических и типовых условий работы и исходных данных. Данное соотношение должно формализовываться в методике тестирования по утверждению между заказчиком и разработчиком, а также устанавливаться в соответствии с техническим заданием на информационной системе. Для информационных систем с высокими показателями надежности могут использоваться форсированные способы тестирования, при которых искусственно вводятся частичные

отказы и повышенные уровни сбоев в аппаратуре и увеличивается интенсивность искажения исходных данных. Значения надежности при ускоренных испытаниях затем должны правильно пересчитываться на нормальные требования эксплуатации. Регистрация отказов и имитация исходных данных может происходить автоматически, но при этом важно предоставить регистрацию условий нарушения работоспособности.

Проверка комфорта эксплуатации и взаимодействия человека с информационной системой предназначена для нахождения трудно формализуемых ошибок представления результирующих и исходных данных. При проверке оценивается удобство представления данных, объем, удобство контроля исходных данных, вводимых непосредственно пользователем, а также результирующих данных, безопасность их использования и удобство изучения. Помимо этого, проверяются динамические параметры отображения и ввода данных в реальном времени. В сложных системах управления основные данные в которых поступают по каналам связи, наибольшее значение имеет проверка принятия решений пользователем в динамике работы системы, особенно в критических ситуациях. Проверка позволяет найти ошибки распределения автоматизируемых функций между ЭВМ и пользователем, а также оценить безопасного решения задач обслуживающим персоналом системы. Часть проверок, которая связана с оценкой комфортного использования документации, может исполняться без вычислительной системы, путем сравнения действий и целей пользователя с содержанием пользовательской документации. При оценке психологического комфорта эксплуатации достаточно большое значение может иметь выбор представленной группы операторов-пользователей. Их критичность, доброжелательность, квалификация могут сильно менять результаты испытаний.

Тестирование качества и удобства подготовки пользовательских версий информационной системы предназначен для поиска ошибок средств и методов настройки ее базовых версий к определенным условиям использования. Множество информационных систем адаптируются перед использованием к операционной среде или к определенным условиям, при которых должны решаться задачи. Для этого могут в автоматизированном режиме подготавливаться данные, которые

будут характеризовать эти требования средств настройки, а также безопасного функционирования адаптированных к всевозможным условиям версий информационной системы. Для контроля средств адаптации производятся специальные тесты, охватывающие наиболее часто встречающиеся режимы применения информационной системы ее пользователями. Проверка адаптированных версий может происходить на основе тестов испытаний на согласие техническому заданию, доработанных по определенной методике для проверки адаптации.

Проверка работы основных версий информационных систем при конфигурациях оборудования применяется для нахождения ошибок, выявляющихся при модификации состава или параметров компонент внешней среды или вычислительной системы. Широко применяемые и серийно выпускаемые базы данных и программы могут работать на вычислительных системах, отличающихся составом оборудования или подключаемых абонентов и их параметрами. Количество возможных конфигураций оборудования может оказаться слишком велико, чтобы все их проверить при подготовке базовой версии информационной системы. Таким образом, большое значение имеют средства и методика подготовки информационной системы к разным конфигурациям оборудования. В состав базовой версии информационной системы вводятся средства, позволяющие адаптировать ее к таким модификациям оборудования. Тесты должны обеспечивать проверку этих средств адаптации во всех возможных режимах комплектации оборудования, а также проверки безопасности адаптированных версий. Поэтому разрабатывается методика подготовки тестов проверки адаптированной версии пользователей, которая применяется настройщиками версий.

Обычно, при проверке необходимо применять имитаторы реальной внешней среды. В таких случаях требования к средствам обеспечения испытаний технологической безопасности информационной системы сходятся к положениям:

– вся информация от имитаторов и объектов внешней среды должна приходить на испытываемую информационную систему в соответствии с обычным ходом процессов в этих системах реального времени;

– интервалы изменения исходных данных в имитаторах должны обеспечивать перекрытие характеристик современных объектов внеш-

ней среды, а также предусматривать возможность расширения этих характеристик с учетом развития информационной системы и прогресса в соответствующих областях техники;

– важно совмещать данные от имитаторов и от реальных объектов внешней среды, замещающих некоторые из них, которые нерационально или невозможно применять при испытаниях в первоначальном виде;

– следует обеспечить контроль, регистрацию и обобщение параметров эталонных, тестовых данных и всех видов аномалий и искажений, поступающих на любом заданном шаге обработки информации и на испытываемую информационную систему в любой момент времени;

– при создании тестовых данных от серии объектов должны учитываться влияния результатов функционирования испытываемой информационной системы по ранее поступавшим данным от тех же объектов с учетом обратных управляющих и информационных связей;

– для всех проверочных данных должны быть подготовлены стандартные реакции информационной системы, с которыми следует сравнивать получаемые во время испытаний результаты;

– нужно обеспечить обобщение и измерение показателей безопасности и качества информационной системы по итогам проведения сеансов испытаний с конкретными целевыми задачами;

– необходимо обеспечить максимальную повторяемость сеансов тестов и испытаний после устранения и нахождения дефектов в работе информационной системы.

Данные требования определяют важность разработки соответствующих проблемно-ориентированных систем, которые способны достаточно полно заменить испытания баз данных и программных продуктов с реально существующими объектами внешней среды. При этом риск испытаний и довольно высокая стоимость с реальными объектами почти всегда оправдывает затраты на данные интегрированные системы в случае, если предстоят проверки критических информационных систем с высокими требованиями к безопасности и надежности работы программ с длинным жизненным циклом и с множеством формирующихся версий.

Сложность адекватного имитирования некоторых объектов среды, главным образом, если в их работе активно принимает участие оператор-пользователь, который не позволяет полностью автоматизировать и сосредоточить всю имитацию тестовых данных на ЭВМ. Поэтому при разработке интегрированных проблемно-ориентированных систем обеспечения проверки безопасности информационных систем требуются аналоги реально существующих объектов среды для создания фрагмента данных и ресурсы вычислительных средств для данных от других объектов. Рациональное сочетание фрагмента реально существующих объектов внешней среды и имитаторов позволяет разрабатывать эффективные моделирующие испытательные стенды с комплексными моделями комплексов объектов, созданных для испытаний безопасности информационных систем реального времени. Данные стенды разрешают генерацию тестов в автоматическом режиме при помощи аналогов реальной аппаратуры и имитаторов на ЭВМ добавлять реальные данные от пользователей, корректирующих и контролирующих работу информационной системы обработки информации. В схеме стандартного МИС можно выделить базовые процессы и компоненты:

– реально существующие объекты внешней среды, т.е. результаты измерений эталонных параметров объектов, данные от аналогов реально существующих объектов, данные натурных экспериментов с объектами, данные с рабочих мест операторов-пользователей;

– моделирующая ЭВМ, т.е. оценка качества работы информационной системы, подготовка документов по результатам испытаний, данные для имитации внешней среды и сеанса испытаний, имитация реальных и эталонных параметров внешней среды, обобщение и синхронизация тестовых данных, оперативная обработка результатов работы информационной системы, обработка и регистрация параметров тестовых данных;

– устройство сопряжения реализующей и технологической ЭВМ;

– реализующая ЭВМ (программы регистрации и оперативной обработки промежуточных данных, испытываемое программное средство).

Важным параметром МИС является их применение в качестве тренажеров для пользователей. Так как безопасность и качество работы информационной системы может зависеть от параметров конкретного человека, принимающего участие в обработке информации, то следует измерять эти параметры. Помимо этого, следует иметь возможность их улучшения до уровня, при котором будет обеспечиваться выполнение заданных требований к системе. Вследствие этого, в процесс испытаний информационной системы входит процесс измерения и тренировки реальной реакции операторов, а также применение стенда для регулярной подготовки операторов-пользователей в процессе эксплуатации и тиражирования информационной системы. Помимо этого, МИС может быть прототипом для создания тренажеров в серийных системах обработки информации и управления.

2.2.2. ОБРАБОТКА РЕЗУЛЬТАТОВ ИСПЫТАНИЙ

Современные проверки систем управления и обработки информации позволяют получить достаточное количество результатов, поэтому достаточно полное их изучение является сложной технической и методической задачей. При избытке контролируемых размеров понижается общее быстродействие ИС и имитаторов в результате определенных затрат времени на регистрацию и контроль, что делает более затруднительным изучение безопасности работы программ в реальном времени. При переходе к многочисленным экспериментам необходимо значительно уменьшать количество изучаемых характеристик и по возможности представлять их в общем виде. В конкретном случае следует стремиться к компромиссу между удобством изучения обобщенных данных и полнотой их, принимая во внимание, что глубокий контроль и обобщение связаны со снижением быстродействия испытываемой системы.

Общим способом получения результатов для экономичного и удобного анализа является иерархическое упорядоченное обрабатывание информации на нескольких уровнях детализации. Таким образом, для контроля системы управления движением в воздухе необходимы координаты объектов, управляющие распоряжения и сообщения от них. При проверке качества сопровождения одного объекта можно

получать корреляционные функции и гистограммы ошибок выходных координат. При расчете контрольных вариантов проверяется информация, которая поступает к диспетчерам, и описывающие работу программных средств и имитаторов признаки. При множественных статистических испытаниях контролируются обобщенные данные (вероятности основных событий, ошибочных и правильных решений, значения средних ошибок для вариантов имитируемой обстановки).

Анализ результатов испытаний информационных систем реального времени может быть поделен на две части: обобщающую и оперативную.

Обобщающая обработка результатов испытаний происходит вне реального времени после завершения экспериментов. Основная задача - расчет интегральных характеристик работы информационной системы. Сложность связана с использованием и получением эталонных данных. Некоторые данные могут быть приобретены от генераторов тестов. При проведении экспериментов с реальными объектами для получения данных используются специальные измерительные комплексы. Совместная обработка и сопоставление экспериментальных данных с эталонными данными тестирования может оказаться непростой задачей.

Оперативная обработка результатов проверки производится по несложным алгоритмам с большой пропускной способностью, которые обеспечивают сохранение реального времени для всего проверяемого комплекса. Основная часть такой обработки результатов связана с замыканием контура обратной связи для имитации работы управляемых объектов. Оперативно следует производить и селекцию некоторых результатов проверки и их первоначальную обработку для сокращения объема сохраняемых результатов.

В оперативную обработку входят манипуляции, связанные с расчетом интегральных данных, которые позволят управлять текущим процессом обработки информации проверяемым программным средством. Стоит выделять, отображать и регистрировать значения параметров, а также ситуации, несущие угрозу безопасности при работе информационной системы.

Объем оперативно отображаемых данных должен быть сокращенным и достаточным для анализа работы информационной систе-

мы. Такие данные должны помогать специалистам, фиксировать условия появления дефектов в работе программ, с учетом автоматической регистрации, которая всегда имеет пробелы в составе отмечаемых характеристик.

2.2.3. МЕТОДЫ ОПРЕДЕЛЕНИЯ ТЕХНОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ КРИТИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ

Экспериментальные способы расчета интегральных показателей безопасности баз данных и программных продуктов в нормальных условиях работы обычно трудно применять при испытаниях на безопасность из-за крупных значений времени наработки на отказ. Достаточная сложность регистрации и выявления нечастых отказов, а также высокая стоимость экспериментов при длительной работе сложных информационных систем приводят к получению малых выборок зарегистрированных отказов на испытаниях. Помимо этого, при таких экспериментах сложно гарантировать полную представительность выборки данных, поскольку для проверки характеризуются конкретными параметрами пользования данной системой на испытаниях.

Для обнаружения тенденции обновления показателей безопасности и надежности их значения обрабатываются статистическими способами. Модификацию этих показателей необходимо связывать во времени с моментами корректировки данных и программ. Изучая корреляцию между процессом изменения программ и значениями надежности, можно определить действия, понижающие надежность и содержащие ошибки. Показатели, получающиеся при этом, позволяют предугадывать число ошибок, которые подлежат исправлению для достижения определенных значений безопасности и надежности в зависимости от длительности проверки. На стадии завершающих проверок при высокой надежности информационной системы может быть полезным применение данных об отказах программ при отладке, что позволяет увеличить достоверность моделей предсказания оценки показателей надежности и проявления ошибок. По совокупности значения моментов выявления отказов математические модели оценки наработки на отказ позволяют определять характеристики модели определенной информационной системы и условий ее откладки

и испытаний. В результате этого может быть оценена наработка на отказ до следующего обнаружения отказа или ошибки. В таком случае, при испытаниях появляется возможность применять опорное значение наработки на отказ, основывающееся на всех результатах обнаружения ошибок при комплексной отладке. Дальнейшее выявление ошибок при испытаниях способствует уточнению характеристик модели, используемой для планирования и предугадывания проверок на безопасность и надежность.

В заключительных испытаниях для правдивого определения безопасности и надежности информационной системы создаются многосуточные и многочасовые прогоны ее в имитированной или реальной среде в атмосфере широкого варьирования с акцентом на стрессовые ситуации исходных данных, побуждающие к действиям угрозы безопасности. Данные прогоны позволяют зафиксировать и измерить показатели безопасности и надежности, а также и степень их соответствия к техническому заданию, а также отметить их в технических условиях на информационную систему.

Поскольку в течение длительного времени интенсивное тестирование программ не ведет к выявлению ошибок, информационная система идет в эксплуатацию. Экспериментальное изучение параметров выявления ошибок в информационных системах позволило оценить скорость выявления ошибок, при котором комплексы программ переходят в регулярную эксплуатацию: число ошибок в день на человека 0,002 – 0,005, т.е. специалисты по отладке проверками или пользователями каждые два месяца эксплуатации информационной системы выявляют примерно одну ошибку. Интенсивность выявления ошибок менее 0,001 ошибок в день на человека, т.е. менее одной ошибки в год на трех-четырех специалистов, принимающих участие в проверке, может быть эталоном отличного качества отладки для информационной системы управления и обработки информации. Во время использования данного критерия обычно берется в расчет календарное время проверки и отладки, включающее продолжительность самой проверки, как для выявления, так и для нахождения ошибок, а также длительность исправления программ и других вспомогательных работ.

Специфическим видом интенсивной проверки является проверка эффективности средств восстановления и контроля вычислительного

процесса, программ и данных. Для этого создаются имитации экстремальных условий работы программ, при которых в наибольшей степени стимулируется функционирование проверяемого средства оперативного восстановления работоспособности и программного рестарта. При таких проверках основная задача заключается в тестировании качества работы средств повышения безопасности и надежности, а оценка интегральных показателей надежности становится второстепенной.

Более сложными являются форсированные проверки эксплуатации ресурсов производительности ЭВМ в реальном времени. При проверке должна быть определена важность качества, безопасности и надежности решения задач от интенсивности получаемой информации. При этом главная задача проверки заключается в вычислении вероятностей, с которыми будет нарушаться соответствие потребностей производительности для решения всей совокупности задач и возможностями ЭВМ. Если возможность нарушения соответствия невелика и можно считать возможным снижение качества за счет задержек и пропусков в обработке сообщений, то делается вывод о соответствии производительности ЭВМ программного средства.

2.3. ТЕХНОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ И ЖИЗНЕННЫЙ ЦИКЛ ИНФОРМАЦИОННЫХ СИСТЕМ

При использовании системы накапливаются дополнительные требования и замечания, а также повышается опыт разработчиков, что стимулирует проведение доработок баз данных и программ. Такие доработки готовятся для очередной версии и вводятся в нее. В самом начале работы с новой версией выявляется определенное число недоработок и ошибок, что вынуждает проектировщиков ограничивать расширение параметров системы и сосредоточивать усилия на увеличении надежности и других показателей качества работы. В результате этого могут появляться две-три промежуточные версии, приближенные по функциям и объему и отличающиеся, в основном, качеством отлаженности. После сокращения количества претензий пользователей к качеству работы, появляется шанс удовлетворить другие требования по развитию функциональных параметров информационных систем.

Существование обратной связи с опозданием может приводить к появлению в характеристиках надежности и безопасности периодической составляющей версий.

На величину диапазона времени между отработанными версиями влияют и запаздывание оформления полной документации, инерционность в передаче версий пользователям и административные планы. В худшем случае реально явление распада системы, когда чрезмерный рост доработок в версии системы делает ее развитие неуправляемым и резко понижает безопасность и эксплуатационные характеристики. Период колебаний объема изменений и соответствующих им параметров качества для информационных систем в разных проектах равен 1–2 годам. Это можно объяснить моделью административной деятельности при сопровождении информационной системы. Для сопровождения коллектив разработчиков имеет ограниченные ресурсы, которые в среднем можно характеризовать бюджетом. Для ввода функций и программных компонент требуются усилия, которые должны увеличиваться в соответствии с повышением сложности информационных систем. Прогрессивному развитию программных продуктов способствует пропорциональное увеличение числа вносимых ошибок. Повышение сложности программ способствует увеличению затрат на борьбу с отставанием корректировок документации, ошибками, повышению квалификации специалистов и так далее. Приходится временно понижать деятельность по развитию программ с той целью, чтобы не выйти за границы допустимых ресурсов. При ограниченных ресурсах реальна предельная критическая сложность сопровождаемых программ, при которой устанавливается динамическое равновесие между вносимыми ошибками и доработками, поэтому не улучшается безопасность и качество информационных систем.

2.4. МЕЖДУНАРОДНЫЕ СТАНДАРТЫ, ПОДДЕРЖИВАЮЩИЕ ИСПЫТАНИЯ ТЕХНОЛОГИЧЕСКОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

Основой развития процесса стандартизации обеспечения технологической безопасности баз данных и программных средств является создание рационального по составу, уровням требований, структуре

комплекса нормативно-технической документации, обеспечивающего нормативный фундамент применения и создания информационных систем. Главной работой в этом направлении служит модернизация в этой области фонда отечественных НТД на основе введения в качестве национальных НТД России международных стандартов.

В таблице 2.1 показаны некоторые, наиболее значимые группы международных стандартов, которые регламентируют:

– качественные показатели программных средств;

– время жизненного цикла и технологический процесс создания критических комплексов программ, способствующих их качеству и предотвращению дефектов;

– проверку программных средств устранения и обнаружения дефектов программ и данных;

– сертификацию программ и испытание для безопасности их функционирования и удостоверения достигнутого качества.

Уровень внедрения и использования международных норм, требований и правил в работающие отечественные НТД на сегодняшний день еще недостаточно высок. В таких условиях создание проектов информатизации без учета рекомендаций и требований международных стандартов ведет к большим растратам технических и финансовых ресурсов из-за увеличения длительности проектирования, производства средств информатизации несовместимости программных и технических средств, необходимости доработки импортируемой продукции.

Технологическая безопасность работы программных баз данных и средств при непредумышленных угрозах поддерживается еще многими стандартами, которые обеспечивают технологию сопровождения и разработки, жизненный цикл, сертификацию баз данных и программ, качество, проверки, тестирование и унификацию интерфейсов с внешней и операционной средой. В таких стандартах предусматривается обеспечение хорошего качества данных и программ в широком смысле слова и отсутствует акцент на технологическую безопасность. Но рекомендации стандартов и тезисы способствуют обеспечению безопасной работы информационных систем.

2.1. Международные стандарты, направленные на обеспечение технологической безопасности

Обозначение стандарта	Название стандарта
ISO 09126:1991. ИТ	Оценка программного продукта. Характеристики качества и руководство по их применению
DOD-STD-2168	Программа обеспечения качества оборонных программных средств
ISO 09000-3:1991	Общее руководство качеством и стандарты по обеспечению качества. Ч. 3: Руководящие указания по применению ISO 09001 при разработке, поставке и обслуживании программного обеспечения
ISO 12207:1995	Процессы жизненного цикла программных средств
DOD-STD 2167 A:1988	Разработка программных средств для систем военного назначения
ISO 09646 – 1-6: 1991. ИТ. ВОС	Методология и основы аттестационного тестирования ВОС
ANSI/IEEE 829 – 1983	Документация при тестировании программ
ANSI/IEEE 1008 – 1986	Тестирование программных модулей и компонент ПС
ANSI/IEEE 1012 – 1986	Планирование проверки (оценки) (verification) и подтверждения достоверности (validation) программных средств
ISO 09126:1991. ИТ	Оценка программного продукта. Характеристики качества и руководство по их применению

2.5. СЕРТИФИКАЦИОННЫЕ ИСПЫТАНИЯ ПРОГРАММНЫХ СРЕДСТВ

2.5.1. ОБЩИЕ СВЕДЕНИЯ О СЕРТИФИКАЦИИ ПРОГРАММНЫХ СРЕДСТВ

До получения пригодного для использования программного продукта оценить его качество можно только вероятностным образом на макроуровне рассмотрения структуры программного комплекса, поэтому возникает потребность в организации определенного этапа при создании программного обеспечения, нужного для подтверждения соответствия качественным показателям реального программного продукта заданным к нему требованиям. Контроль исполнения этих требований должен происходить с учетом условий применения при больших нагрузках и проверке всех установленных режимов. В рамках разработки современных информационных технологий решение задач проверки ПО и получения документального подтверждения требуемых показателей качества программ объединяется в рамках процесса сертификации.

Сертификация программ представляет собой процесс проверки программ в загруженных режимах использования, подтверждающий соразмерность качественных показателей программного продукта требованиям, указанным в нормативно-технических документах на этот продукт и дающий документальную гарантию применения программного средства при следовании заданным ограничениям.

Сертификация программного продукта компьютерных систем возможна при следовании следующим условиям:

– создании шкалы качественных показателей с учетом специфики применения программных средств и набора их функциональных параметров;

– каталогизации программ как объекта сертификации на основе опыта их работы;

– разработки специализированного центра сертификации, исполняющего роль «третейской» организации контроля качества;

– создании нормативно-технической базы, регламентирующей сертификацию программных продуктов;

– создании эталонов программных средств, которые будут удовлетворять требованиям технических заданий на создание разнотипных программных комплексов;

– создании специальной технологии проверок, определяющей объем и содержание сертификационных проверок;

– создании комплекса проверочного программного обеспечения для широкого диапазона программных продуктов.

При сертификации сложного программного обеспечения следует выделить аспекты: технологический и методический. Технологический аспект связан с автоматизацией процесса применения методического аппарата, а методический – с созданием комплекса методик сертификации программного обеспечения с учетом специфики его работы.

Необходимо отметить, что по некоторым оценкам до 70% затрат на внедрение и разработку сложных программных комплексов приходится на исполнение процесса их сертификации. Значительная доля этих затрат относится к организации моделирующих средств, аппаратно-программной платформы и тестового обеспечения стенда сертификации.

Помимо того, важным вопросом разработки качественных программных продуктов является предоставление технологической безопасности программного обеспечения на этапе сертификационных стендовых испытаний. Неполный уровень развития современных информационных технологий разработки программного обеспечения, преимущественное использование зарубежных инструментальных средств и использование разработчиками лишь средств защиты от непреднамеренных изъянов обуславливают существенные, принципиально новые модификации технологии формирования программ в этих условиях. Следовательно, одной из задач сертификации на современном уровне формирования информационных технологий становится поиск преднамеренных программных недостатков.

Технологическая безопасность на стадии сертификационных проверок определяется усилением мер контроля, поскольку в настоящее время предполагается, что вероятность введения закладок на заключительных фазах создания программ больше, чем на исходных фазах в связи с уменьшением вероятности их нахождения при последова-

тельном технологическом контроле, так как с этой конечной процедурой тестового контроля и проверки программ должна быть сертификация программного обеспечения по запросам безопасности с выпуском сертификата на соответствие этого продукта требованиям технического задания. В обстановке существующих технологий разработки программного обеспечения его сертификация выступает быстрореализуемым способом «фильтрации» систем от программных средств низкого качества и не отвечающих условиям безопасности.

Сертификационные испытания программ, в том числе защищенных, и программных продуктов контроля защищенности происходят в отраслевых и государственных сертификационных центрах.

Право на выполнение сертификационных проверок защищенных программных средств вычислительной техники предоставляется Федеральной службе по техническому и экспортному контролю РФ по согласованию с Госстандартом России предприятиям-разработчикам защищенных средств вычислительной техники и специализированным организациям ведомств, которые разрабатывают защищенные средства вычислительной техники.

В согласовании с «Положением о сертификации средств защиты информации» и «Положением о сертификации продукции по требованиям безопасности информации» по результатам сертификационных проверок оформляется акт, а разработчику предоставляется сертификат, дающий право на распространение и применение этих средств, как защищенных и заверенных в ФСТЭК России.

Средства, которые получили сертификат, входят в номенклатуру защищенных средств вычислительной техники.

Созданные программные продукты после приемки регистрируются в специализированном Государственном фонде алгоритмов и программ.

2.5.2. ПРАКТИЧЕСКИЕ АСПЕКТЫ СЕРТИФИКАЦИОННЫХ ИСПЫТАНИЙ ПО

Сертификация на недостаток недекларированных возможностей направлена на специализированное программное обеспечение, созданное для защиты информации ограниченного доступа. На сегодняшний

момент общий объем такого программного обеспечения достаточно велик. Это связано с требованием защиты самой разнообразной информации при ее создании, транспортировке и обработке. Несомненно, что вопрос защиты информации насущный для построения любого вида собственности, так как потенциальными пользователями такого специализированного программного обеспечения являются как коммерческие, так и государственные структуры. Помимо этого, следует брать в расчет то, что сертификационные проверки, о которых будет сказано, законодательно не используются в программном обеспечении средств криптографической защиты информации.

Сертификационные проверки на нехватку недекларированных возможностей подразумевают глубокое изучение программного обеспечения и согласованы с изучением исходного и исполняемого кода с целью установления факта наличия или отсутствия в некотором программном решении возможностей, не описанных разработчиком.

Основой такого изучения являются общие принципы анализа программ с учетом критериев, связанных с информационной безопасностью. Практические и теоретические труды в этой области известны давно. Но, переход этих мероприятий в число управляемых Российским государством в информационной безопасности произвелся в 1999 году при появлении в системе сертификации Гостехкомиссии России нового руководящего документа «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей».

Документом предусмотрено четыре уровня контроля программного обеспечения, отличающихся объемом, условиями и глубиной проведения проверок. Существование нескольких уровней контроля программного обеспечения при проведении проверок на отсутствие недекларированных возможностей регламентирует степень конфиденциальности информации, защита которой производится программным обеспечением, проверенным по какому-либо уровню. С другой стороны, разные уровни управления позволяют разделять степень вероятности отсутствия в изучаемом программном обеспечении этих самых возможностей.

Самым нижним уровнем контроля является четвертый. Четвертый уровень предусмотрен для проверки программного обеспечения, применяемого при защите конфиденциальной информации. Для программного обеспечения, применяемого для защиты государственной тайны, при проведении проверок должен быть гарантирован уровень контроля выше или равный третьему. В соответствии с условиями законодательства для программного обеспечения, предназначенного для изучения информации, включающей государственную тайну, проведение проверок по уровням контроля отсутствия недекларированных возможностей обязательно.

Не вдаваясь в подробности относительно требований к уровням контроля, рассмотрим отличия в управлении по третьему и четвертому уровням контроля.

Существенным отличием между отмеченными уровнями контроля является то, что четвертый уровень подразумевает мероприятия только по статическому анализу программного обеспечения, а при третьем уровне контроля выполняются мероприятия по статическому и динамическому изучению. В терминах документа Гостехкомиссии России под статическим анализом подразумевается совокупность способов контроля соответствия или несоответствия декларированных и реализованных в документах характеристик программного обеспечения. Эти методы основываются на декомпозиции и изучении исходных текстов программных продуктов. Под динамическим изучением понимается совокупность способов контроля соответствия или несоответствия созданных и декларированных в документации характеристик программного обеспечения и методы, основанные на идентификации реально существующих маршрутов выполнения объектов с дальнейшим сопоставлением потенциальным маршрутам исполнения, созданным в процессе статистического анализа. Иными словами, во время статистического анализа получается вероятностная картина поведения программного обеспечения и проверяется, как это назначено разработчиком программного обеспечения. Во время динамического анализа проявляется возможность проверки поведения программного обеспечения в объявленных разработчиком режимах работы и производится

сопоставления вероятностного и реального поведения программного обеспечения.

Отметим, что процесс статического анализа в части технологических манипуляций отличен для третьего и четвертого уровней контроля. Различие состоит в объеме регламентированных технологических манипуляций для каждого отдельно взятого уровня.

Данное утверждение вовсе не обозначает несерьезность статистического анализа для четвертого уровня. Решение задании поиска недекларированных возможностей происходит на регламентированном подмножестве технологических манипуляций.

Происходит контроль соответствия кодов программного обеспечения его загрузочному коду. Также контролируется полнота и избыточность представленного кода, происходят мероприятия по управлению первоначального состояния программного обеспечения, связанные с отметкой первоначального состояния при помощи механизмов суммирования, и сравнение результатов отметки с приведенным в документах, производится экспертиза представленной создателем эксплуатационной и технологической документации программного обеспечения.

В процессе изучения происходит поиск инженерных паролей, дающих возможность обходить механизмы защиты. Производится контроль корректности сборки кода из исходного с целью исключения возможности доступа к отладочной информации в процессе штатной работы программного обеспечения.

Помимо этого, мероприятия по статическому анализу программного обеспечения для третьего уровня контроля дополнительно предусматривают произведение анализа отсутствия избыточности и полноты представленного исходного кода на уровне функциональных объектов, осуществление управления связями функциональных объектов по администрированию и информации, исполнение контроля информационных объектов различных видов, создание перечня возможных маршрутов исполнения функциональных объектов.

Разработчика программного обеспечения, которое объявляется как средство защиты конфиденциальной информации, никто не вынуждает проводить сертификацию по недекларированным возможностям, она носит добровольную основу. В такой сертификации есть

плюсы, о чем будет сказано далее. Но, принимая во внимание более полные критерии статического изучения для третьего уровня контроля, разработчик программного обеспечения, которое не создано для защиты информации, отнесенной к гостайне, ровным счетом ничего не теряет, если сертифицирует это программное обеспечение по третьему уровню. А получает такой разработчик возможность выхода на государственный рынок, так как оценка наличия в его программном обеспечении недекларированных возможностей хоть и останется вероятностной, но будет строже. В любом случае фигурирование разработчика программного обеспечения в процессе сертификации по контролю отсутствия недекларированных возможностей будет актуальным, так как актуальным является сам вопрос программных закладок.

Рассмотрим предпосылки повышения популярности проверок на отсутствие недекларированных возможностей:

– существенно повысился уровень возможностей разных средств защиты информации, что сделало малоперспективными и дорогостоящими мероприятия по взлому систем защиты без знаний об их принципах работы и построении;

– неизмеримо увеличилась ценность и значимость защищаемой информации разного уровня конфиденциальности;

– пользователь информации вник в суть проблемы и стал сторониться попыток несанкционированного доступа к информации со стороны хакеров и со стороны разработчиков программных продуктов, фискальных органов и тому подобных.

Для приобретения несанкционированного доступа к информации хакеру проще использовать заранее подготовленные закладки в программе, а потребитель хочет получить гарантии отсутствия доступа к информации.

Необходимо наличие у поставщика или производителя программного обеспечения сертификата, который будет подтверждать отсутствие программных закладок. К сертифицированной продукции повышается доверие старых заказчиков, крупных потребителей.

Технологические преимущества приобретает непосредственный создатель сертифицированного программного обеспечения. Разработчик и компания реализует самостоятельно программное обеспечение на рынке потребителя.

Суть преимуществ обусловлена спецификой произведения сертификационных испытаний такого вида. Присущие контролю отсутствия недекларированных возможностей технологические операции, такие, как идентификация объекта сертификации, мониторинг исходной инсталляции программного обеспечения, мониторинг сборки исполняемого кода из начального позволяют получить полный объем полезной информации.

Ранее упоминалось, что существует возможность четкого соотнесения исполняемого и исходного кода программного обеспечения, устранения и выявления избыточности исходного кода, однозначного определения действий программного обеспечения в процессе начальной установки и деинсталляции по отношению к системным областям системы ЭВМ.

Как уже упоминалось ранее, возможно четко соотнести исходный и исполняемый код ПО, выявить и устранить избыточность представленного. Результаты проверок могут быть применены разработчиком программ для проведения более точного анализа созданного продукта, реализации и планирования корректирующих воздействий на программное обеспечение в части улучшения процессов его сопровождения и создания.

Присущая для высоких уровней контроля по недостатку недекларированных возможностей технологическая процедура по динамическому изучению программного обеспечения предусматривает фактическую ее проверку. Такая проверка является углубленной, учитывающей не только возможности изучаемого программного обеспечения, но и его структурные и технологические особенности. Это объясняется важностью инициирования отработки программным обеспечением логических маршрутов ее исполнения, которые были рассчитаны как возможные на шаге статического анализа исходного кода. Нет сомнений в полезности для разработчика результатов проверки, выполненной какой-либо независимой организацией. В дополнение разработчик программного обеспечения получает исчерпывающие данные о степени соотношения представленной на сертификационные проверки программной документации условиям соответствующей нормативной документации. Очевидно, что такая информация важна для разработчика

при сопровождении своего изделия, создании новых программных продуктов или версий.

Следующим важным моментом является процесс проверки по контролю отсутствия недекларированных возможностей объективно предопределяет постоянным контактом разработчика и пользователя, что часто позволяет разработчику оперативно модифицировать потребительские параметры программного продукта в процессе проверки. В интересах разработчика проверки могут быть исполнены на его производственной базе.

К числу маркетинговых или потребительских можно отнести следующие преимущества:

– сертификат с некоторой степенью вероятности, зависящей от уровня произведенного контроля, подтверждающий тот факт, что в проверенном программном обеспечении нет очевидных программных конструкций, применение которых подразумевает возможность незаконного доступа, нарушения целостности информации;

– опровергается отсутствие также являющихся недекларированными, так называемых критичных программных конструкций, которые способны в некоторых условиях спровоцировать нештатные в отношении защищаемых данных;

– при выполнении проверок по более высшим уровням контроля, результирующим сертификатом доказывается способность программного обеспечения реализовывать свои продекларированные допустимости;

– по результатам испытаний программное обеспечение приобретает определенный идентификационный признак – отмеченные контрольные суммы исполняемых и исходных файлов, разрешающие осуществлять мероприятия по контролю целостности программного обеспечения на этапах его создания, распространения и работы;

– критерии оценки программного обеспечения регламентированы системой сертификации ФСТЭК России, которая изначально была направлена на защиту информации, отнесенной к государственной тайне, а, следовательно, данная система критериев достаточно обоснована и серьезна и может быть применена для оценки защитных пара-

метров программного обеспечения, работающих не только с конфиденциальной, но и секретной информацией;

– за транспортировку пользователю сертифицированного программного обеспечения отвечает не только его создатель, но и проводившая проверку сертификационная организация, которой нормативными документами внесены соответствующие контрольные возможности;

– дополнительная возможность, которую готова дать испытательная организация – подтверждение установки сертифицированного программного обеспечения соответствующим актом на объектах пользователей.

Существует еще один критерий данной темы. Данный критерий не может быть подвержен классификацией преимуществ, предложенной выше, однако, являясь реальностью, должен быть принят в расчет всеми заинтересованными сторонами. Известно, что работа сертификата соответствия в системе сертификации ФСТЭК России ограничена во времени. Также общеизвестна процедура продления и подтверждения работы сертификата, основанная на изучении соответствия сертифицированных параметров вновь отдаваемого на сертификацию продукта по отношению к таким же свойствам сертифицированного эталона. При наличии нового документа по контролю отсутствия недекларированных возможностей в программном обеспечении возникает определенная ошибка, выраженная в том, что ранее сертифицированные средства защиты информации, имеющие сертификаты соответствия, которые позволяют применять эти средства для защиты государственной тайны, после завершения работы сертификатов не могут быть сертифицированы вторично в том же качестве без расчета требований нового руководящего документа. Либо такая вторичная сертификация, без проверки программного обеспечения на отсутствие недекларированных возможностей, по факту окончания понизит гриф секретности информации, обрабатываемой с помощью пересертифицированного средства защиты, до уровня конфиденциальной, без возможности обработки информации, связанной с государственной тайной. Возможно, что существование сертификата по некоторому уровню контроля отсутствия недекларированных возможностей в данной ситу-

ации все же является большим преимуществом для программных продуктов, претендующих и дальше позиционироваться на государственном рынке средств защиты информации.

Все предшествующие рассуждения являются оценкой работы испытательных лабораторий, получившей подтверждение представителями множества компаний, в чьих интересах проводились проверки по контролю отсутствия недекларированных возможностей.

Существуют примеры, когда результаты проведенных проверок позволяли создателю программного обеспечения улучшить качество своих технологических решений, выходить на новые уровни рынка.

Определенно показательны некоторые из последних испытаний – сертификация программного обеспечения цифровой системы связи «*CORAL-3*» – первые в России испытания по контролю недекларированных возможностей программных продуктов автоматической телефонной станции (АТС) зарубежного производства. Объективными предпосылками этих испытаний являются:

1) отсутствие определяющего документа с условиями по безопасности в АТС;

2) возможная востребованность этих систем связи в силовых и государственных структурах;

3) важность наличия сертификата по безопасности информации для продвижения продукции на рынок. Результатами и преимуществами, полученными в результате сертификации, являются:

1) присутствие сертификата ФСТЭК России с возможностью участия в обработке конфиденциальной информации;

2) формальное удовлетворение условий заинтересованных государственных структур и силовых ведомств;

3) понимание разработчиком и пользователем технологических мероприятий, достигнутое в процессе испытаний, которые делают возможными испытания по более высокому уровню контроля, сертификат, по результатам которого разрешит использование данной продукции при обработке засекреченной информации.

Применяя сертифицированное по контролю отсутствия недекларированных возможностей программное обеспечение, пользователь получает средства, которые с некоторой степенью вероятности делают две вещи:

– гарантированно и четко исполняют функции по защите информации;

– не имеют в наличии встроенных механизмов, разрешающих нанести информации вред.

Контрольные вопросы

1. Назовите и поясните виды тестирования для определения технологической безопасности информационных систем.
2. Прямые экспериментальные и статистические методы определения технологической безопасности критических информационных систем.
3. Сформулируйте цель сертификации ПО и перечислите решаемые в ходе сертификации задачи.
4. Кто уполномочен производить сертификационные испытания ПО?
5. Поясните четыре уровня контроля безопасности ПО.

3. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

3.1. ВОЗМОЖНОСТИ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Обнаружение вторжений (атак) – это процесс оценки возникающих в сети подозрительных событий.

Атака – это действия злоумышленника, представляющие собой преднамеренную попытку обойти службы безопасности и нарушить политику безопасности системы. Со стороны нейтрального наблюдателя атака может быть безуспешной (или неудачной) и успешной (т.е. удачной). Успешно проведенную атаку принято называть вторжением. Вторжение – это несанкционированный доступ к информационной системе при применении минующих систему или нарушающих политику безопасности защиты действий.

Основными причинами удачного проведения удаленных атак на распределенные вычислительные системы являются:

– уязвимости в процедурах идентификации и аутентификации объектов и субъектов;

– отсутствие полных сведений о системных объектах;

– не используется криптозащита сообщений;

– нет возможности контролировать маршруты сообщений;

– отсутствует проверка и поддержка виртуальных каналов связи между системными объектами;

– нет предоставленного канала связи между объектами системы.

Система обнаружения вторжений (СОВ) – это аппаратное или программное средство, специализированное на обнаружении случаев несанкционированного управления, например, через Интернет, либо неавторизованного доступа в компьютерную систему или сеть. Существует также английский термин системы обнаружения вторжений – *Intrusion Detection System* (*IDS*).

СОВ создает добавочный уровень защиты безопасности компьютерных систем: локализацию места воздействия нарушителя; неотказуемость отправки/доставки сообщений (при регистрации событий

в журнале); сообщение о нарушениях и восстановлении нарушенного процесса функционирования (мониторинг консолью сенсоров/серверов).

Применение СОВ позволяет достичь следующих целей:

– документально зафиксировать имеющиеся угрозы;

– получить и систематизировать сведения о уже происходивших ранее вторжениях для воссоздания и исправления причин, вызвавших эти проникновения;

– определить месторасположение относительно сети источника атаки (внутренние или внешние), что существенно при принятии решений о месторасположении ресурсов в сети;

– выявить сетевую атаку или вторжение;

– прогнозировать вероятные атаки в будущем и выявлять уязвимости для предотвращения их дальнейшего использования;

– установить контроль качества администрирования относительно вопроса безопасности.

Для обнаружения отдельных видов деятельности, нарушающей политику безопасности системы, обычно применяются СОВ. Так, к представленной активности можно отнести сетевые атаки по отношению к уязвимым сервисам, ориентированным на увеличение прерогатив атаки, неавторизованный доступ к значимым файлам, а также действия вредоносного программного обеспечения (таких как компьютерные вирусы, трояны и сетевые черви).

Характерной является возможность интеграции СОВ с межсетевыми экранами и антивирусными системами, когда в последние добавляются функции активного аудита.

Любая СОВ должна обладать следующими свойствами:

– выполняться непрерывно, в фоновом режиме;

– быть отказоустойчивой к отказам и сбоям;

– быть неподверженной атакам;

– создавать минимальную нагрузку на систему;

– быть приспособленной для сред с шифрованием;

– уметь адаптироваться к изменениям (в сети, приложениях, устройствах).

Далее рассмотрим структуру типовой СОВ.

3.2. АРХИТЕКТУРА СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Большая часть современных концепций обнаружения вторжений строится на базе архитектуры «клиент-сервер». В этих системах можно выделить несколько ключевых компонентов:

– подсистему сбора данных, которая используется для собирания информации о деятельности защищаемой системы. Она включает в себя набор датчиков;

– подсистему анализа, которая включает в себя анализаторы и позволяет выявлять инциденты в защищаемой системе;

– модуль представления данных, который позволяет пользователю наблюдать за состоянием защищаемой системы и обнаруженными атаками.

Концепция обнаружения вторжений также включает хранилище для хранения обработанных и первичных событий, а также консоль управления, которая позволяет настраивать систему обнаружения вторжений.

Типовое устройство современных систем обнаружения вторжений изображено на рис. 3.1.

Подсистема сбора информации используется для обработки и накопления изначальной информации о работе защищаемой системы. Датчики, которые являются автономными модулями, применяются для сбора данных. Количество применяемых датчиков зависит от особенностей защищаемой системы и может быть различным. В системе обнаружения вторжений датчики классифицируются по типу собираемой информации. Существует четыре типа датчиков, соответствующих стандартной структуре информационных систем:

– датчики приложений, собирающие информацию о работе программного обеспечения защищаемой системы;

– датчики хоста, фокусирующиеся на информации о функционировании рабочей станции защищаемой системы;

– датчики сети, собирающие данные для оценки сетевого трафика;

– межсетевые датчики, содержащие характеристики данных, которые передаются между сетями.

В системе обнаружения вторжений используются разные комбинации этих датчиков.

Для уменьшения объема информации и упрощения работы остальных компонентов системы возможно применение фильтрации информации на уровне датчиков. Данные от датчиков передаются в центральный узел, который осуществляет фильтрацию, сохраняет информацию в базе данных и направляет ее на анализ.

Подсистема анализа (обнаружения) играет важную роль в защищаемой системе, так как она отвечает за обнаружение вторжений и атак. Структурно эта система включает один или более анализаторов (модулей анализа). Наличие нескольких анализаторов необходимо для улучшения производительности обнаружения.

Каждый анализатор выполняет определение конкретных видов атак и вторжений. Входными данными для анализатора являются данные из подсистемы сбора информации или от других анализаторов. Итоговая информация о состоянии защищаемой системы считается результатом работы подсистемы. Система обнаружения вторжений должна быть способна ясно и понятно объяснить, почему возникла тревога, насколько серьезна ситуация и какие рекомендации по дальнейшим действиям. Если пользователю требуется сделать выбор, то ему предоставляется несколько вариантов в меню, но не решение концептуальных проблем.

Подсистема представления данных (пользовательский интерфейс) необходима для информирования заинтересованных лиц о состоянии защищаемой системы. Эта подсистема позволяет пользователям системы обнаружения атак наблюдать за состоянием защищаемой системы. В некоторых случаях предусмотрено наличие групп пользователей, каждая из которых отвечает за контроль определенной подсистемы защищаемой системы. Это позволяет использовать полномочия, групповые политики безопасности и разграничение доступа при применении этих методов обнаружения атак.

Кроме того, целесообразно интегрировать в систему обнаружения вторжений модуля автоматического реагирования на обнаруженные атаки и вторжения. Вариативность реагирования системы возможно от уведомления и запоминания до активного реагирования.

Таким образом, одним из возможных способов реализации системы обнаружения атак является система, устройство которой представлено на рис. 3.2.

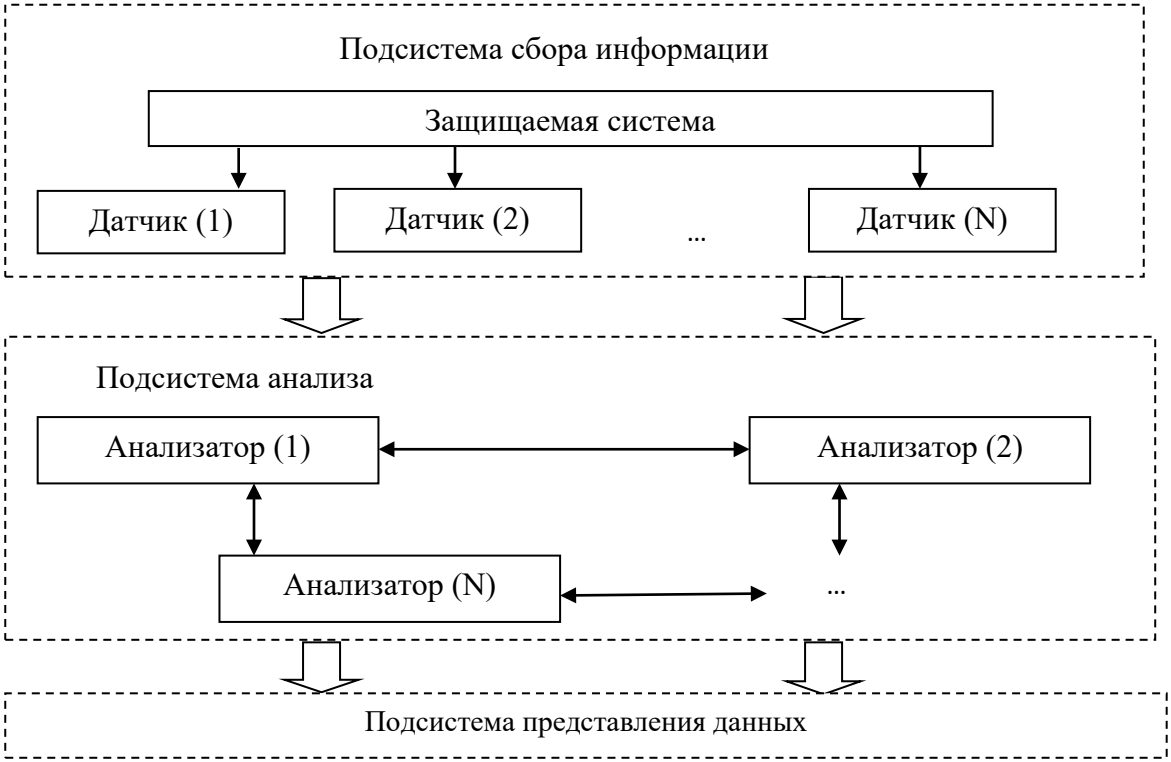

Рис. 3.1. Устройство современных систем обнаружения вторжений

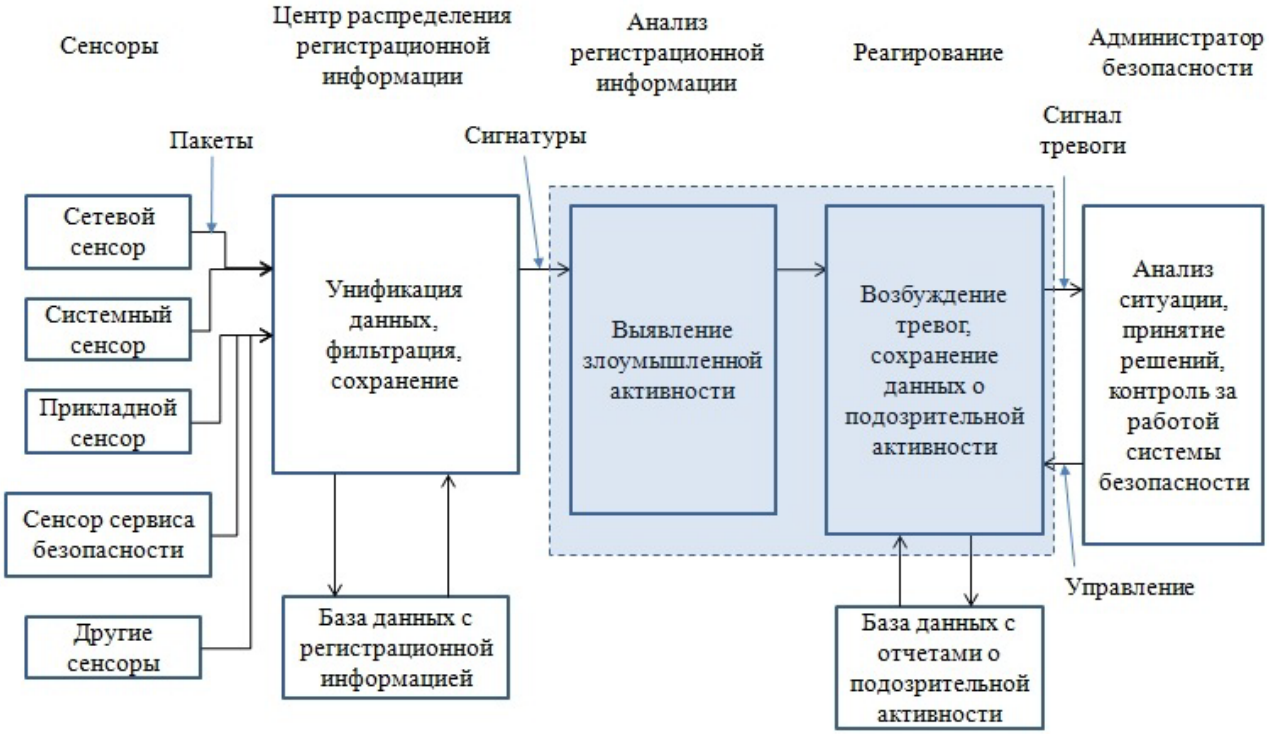

Рис. 3.2. Устройство варианта реализации СОВ

Существует структура информации, которая собирается сенсорами и передается в центр распределения регистрационной информации. В этом центре определяется подозрительная деятельность и передается в центр реагирования, который постоянно обменивается информацией с администратором безопасности на основе анализа регистрационных данных.

Система обнаружения вторжений использует различные методы классификации, в зависимости от типа и расположения сенсоров, а также методов, применяемых для раскрытия подозрительной деятельности подсистемой анализа. В большинстве обычных систем обнаружения вторжений все элементы представлены в виде одного модуля или устройства.

3.3. КЛАССИФИКАЦИЯ СИСТЕМ ОБНАРУЖЕНИЙ ВТОРЖЕНИЙ

Классификация систем обнаружения вторжений возможна по ряду признаков, таких как:

– методы обнаружения;

– особенности защищаемой системы;

– стратегии управления;

– специфичность источников данных;

– время анализа события;

– реакция системы.

Обобщенная классификация системы обнаружения вторжений представлена на рис. 3.3.

Существуют различные методологии обнаружения нарушений безопасности, включающие системы, основанные на выявлении сигнатур (*signaturedetection, misusedetection*), системы, основанные на выявлении аномалий (*anomalydetection*), а также гибридные системы, которые сочетают в себе оба метода.

Защищаемая система может быть классифицирована как хостовая, сетевая или гибридная. При анализе на уровне хоста систему можно подразделить на операционную систему, базу данных и приложение, каждое из которых определяет атаки, соответствующие конкретным компонентам.

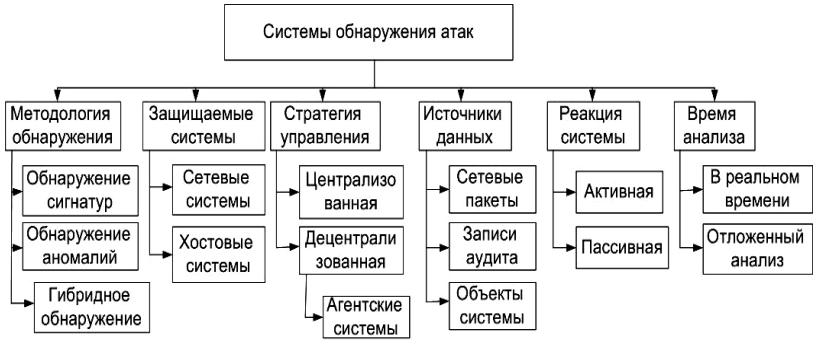

Рис. 3.3. Классификация СОВ

Сетевые системы обнаружения нарушений работают путем получения и исследования сетевых пакетов. Сенсоры этих систем могут быть размещены на различных сетевых компонентах.

Хостовые системы обнаружения нарушений выполняют анализ активности на отдельном компьютере. Для анализа таких систем требуется более широкий набор характеристик. Кроме того, эти системы могут просто реагировать на обнаруженные атаки.

Хостовые системы также могут наблюдать события, связанные с приложением, и исследовать входящие пакеты на хосте. Хостовые системы, которые анализируют входящие сетевые пакеты, считаются гибридными, так как применяют оба подхода одновременно.

В составе систем обнаружения вторжений можно выделить два основных типа: централизованные и децентрализованные. Почти все существующие системы, будь то хостовые или сетевые, имеют монолитную архитектуру, где используются все функции системы: получение информации, ее анализ и принятие решений, включая сигналы тревоги. Сложность современных атак, связанных с различными аспектами безопасности сети, устройств и хостов, а также высокая стоимость отдельных систем обнаружения вторжений привели к необходимости разработки распределенных систем атак.

Системы обнаружения вторжений могут быть классифицированы по источникам данных, используемым в них. Входной информацией для них могут быть лог-файлы системы, сетевые пакеты или результаты аудита.

Системы обнаружения вторжений также могут быть классифицированы по времени анализа, на постоянно функционирующие и периодические. Это разделение определяет два разных подхода к обнаружению, обычно называемые режимом реальным временем и отложенным режимом.

В случае обработки в режиме реального времени производится непрерывная верификация событий системы для хостовых систем и анализ сетевых пакетов для сетевых систем обнаружения вторжений. Из-за вычислительных сложностей, используемые алгоритмы в сетевых системах обнаружения вторжений обычно ограничиваются своей эффективностью и быстродействием, что зачастую требует использования более простых алгоритмов.

Системы обнаружения атак работают только в режиме реального времени, в то время как системы обнаружения вторжений могут работать и в режиме реального времени, и в отложенном режиме.

В зависимости от реакции системы, они могут быть пассивными, просто генерирующими сигналы тревоги, или активными, выполняющими обнаружение и реакцию на атаки системы обнаружения вторжений. В качестве реакции могут быть приняты меры по обнаружению уязвимостей в программном обеспечении перед атакой или блокировке атакуемых служб.

Пассивные системы обнаружения вторжений могут отправлять сигналы тревоги на консоль администратора, отправлять уведомления по электронной почте или даже звонить на указанный мобильный телефон. Политика управления определяет, каким образом компоненты системы обнаружения вторжений могут быть управляемы, а также управление исходными и конечными данными.

Политика управления позволяет определять, каким способом можно использовать компоненты системы обнаружения вторжений, их исходные данные и полученные результаты.

В сети необходимо применять следующие связи:

– связи, необходимые для трансляции отчетов СОВ (формируются между сенсорами как сетевого мониторинга, так и мониторинга хоста, и центральной консолью системы обнаружения вторжений);

– связи для контроля хостов и сетей;

– связи для исполнения ответов системы обнаружения вторжений.

При централизованных стратегиях управления обнаружением, анализом, обработкой и отчетностью управляют напрямую из единого центра управления. В описанном случае существует только одна связанная со всеми размещенными в сети сенсорами консоль системы обнаружения вторжений.

При частично распределенном управлении обнаружение и анализ осуществляются с локально управляемого узла, а отчетность – в один или несколько центров.

При полностью распределенном управлении обнаружение, анализ, обработка осуществляется применением подхода, основанного на агентах, когда решение принимается производятся в точке анализа.

3.4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

3.4.1. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ЗАЩИЩАЮЩИЕ СЕГМЕНТ СЕТИ

Существуют два основных типа систем обнаружения вторжений: защищающие системный сегмент сети и защищающие отдельные системные хосты.

Первые анализируют сетевые пакеты, а вторые используют данные, основанные либо на операционной системе, либо на приложении. Рассмотрим преимущества и недостатки каждого из этих типов систем.

Основные коммерческие системы обнаружения вторжений – это сетевые системы. Они устанавливают тип атаки, анализируя сетевые пакеты. Сетевые системы обнаружения вторжений (СОВ) могут прослушивать сетевой сегмент и защищать данные хосты, входящие в этот сегмент. Часто сетевые СОВ состоят из нескольких сенсоров, размещенных в разных местах сети. Эти сенсоры анализируют локальный сетевой трафик и создают отчеты об атаках для центральной управляющей консоли. Большинство сенсоров работают в «невидимом» режиме, что затрудняет обнаружение их наличия и местоположения атакующими.

Преимущества сетевых СОВ:

– развертывание сетевых СОВ не влияет на эффективность сети, так как они являются пассивными устройствами, не вмешивающимися в нормальное функционирование сети;

– сетевые СОВ могут быть практически неуязвимыми или невидимыми для атакующих;

– несколько хорошо расположенных сетевых СОВ могут мониторить большую сеть.

Недостатки сетевых СОВ:

– сетевые СОВ могут иметь проблему определения типа атак, включающих фрагментированные пакеты. Это может привести к нестабильной работе системы обнаружения вторжений и использованию этой уязвимости атакующими;

– большинство преимуществ сетевых СОВ не применимы к современным сетям, основанным на коммутаторах. Коммутаторы разделяют сети на множество небольших сегментов и не обеспечивают универсального мониторинга портов, что ограничивает возможности мониторинга сенсором сетевого СОВ;

– сетевые СОВ не могут анализировать зашифрованные данные, что становится проблемой при использовании *VPN*;

– большинство сетевых СОВ могут только установить начало атаки, но не могут определить, была ли атака успешной. Это означает, что администратору необходимо вручную проверять каждый атакованный хост для определения факта вторжения;

– сетевые СОВ имеют проблему обработки всех пакетов в загруженной сети и могут упустить обнаружение начавшейся атаки при большом трафике. Кроме того, вычислительные ресурсы могут быть ограничены, что снижает эффективность обнаружения.

3.4.2. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ЗАЩИЩАЮЩИЕ ОТДЕЛЬНЫЙ ХОСТ

Хостовые или системные системы обнаружения вторжений (СОВ) работают с данными, накопленными внутри одного компьютера. Защищающие отдельные приложения СОВ являются частью хостовых СОВ. Это позволяет системным СОВ исследовать работу с высокой достоверностью и точностью, анализируя только процессы и пользователей, связанные с конкретной атакой на операционную систему.

Системные СОВ имеют прямой доступ к данным системы, файлам и процессам, которые могут быть целью атаки, поэтому они

способны «видеть» результаты атаки. Системные СОВ используют информационные источники двух видов: аудит операционной системы и системные логи.

Аудит операционной системы является более подробным и лучше защищенным, так как формируется на уровне ядра. Системные логи менее объемные и проще для представления.

Некоторые хостовые СОВ созданы для поддержки централизованной инфраструктуры управления и получения отчетов, что позволяет отслеживать множество хостов через единственную консоль управления. Другие генерируют сообщения, совместимые с системами сетевого управления.

Преимущества хостовых СОВ:

– они могут обнаружить атаки, которые не видны сетевым СОВ, так как наблюдают события локально на хосте;

– они могут функционировать в среде с зашифрованным сетевым трафиком, так как данные формируются до шифрования или после расшифровки на хосте;

– присутствие коммутаторов в сети не влияет на работу хостовых СОВ;

– когда хостовые СОВ используют аудит операционной системы, они могут обнаружить троянские программы и другие атаки, нарушающие целостность программного обеспечения.

Недостатки хостовых СОВ:

– они требуют сложного управления, так как данные должны быть сконфигурированы и управляемы для каждого просматриваемого хоста;

– так как источники информации для системных СОВ находятся на хосте, который может быть целью атаки, то сама СОВ может быть атакована и отключена;

– хостовые СОВ не могут полностью обнаружить сканирование сети или другие подобные исследования, если целью является вся сеть, так как они следят только за сетевыми пакетами, полученными конкретным хостом;

– хостовые СОВ могут быть заблокированы определенными DoS-атаками;

– использование аудита операционной системы в качестве источника информации может привести к большому объему данных, требующих дополнительного хранения;

– хостовые СОВ используют вычислительные ресурсы хостов, за которыми они следят, что может влиять на производительность системы.

На сегодняшний день имеются два основных подхода к рассмотрению событий: установление сигнатуры и выявление аномалии.

3.4.3. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА ОСНОВЕ АНАЛИЗА СИГНАТУР

Первоначально анализ сигнатур был использован для обнаружения вторжений. Этот метод основывается на сравнении последовательности входящего пакета с заданными сигнатурами. Каждый байт в пакете проверяется на соответствие сигнатуре, которая является уникальной строкой программы и указывает на характер вредоносного трафика. Если происходит совпадение, то срабатывает уведомление.

Системы анализа сигнатур обладают рядом преимуществ. Во-первых, они быстрые, так как анализ всего пакета является относительно несложной задачей. Правила для сопоставления могут быть легко созданы, поняты и настроены. Поддержка всего компьютерного сообщества позволяет быстро создавать новые сигнатуры для обнаружения новых угроз. Эти системы превосходят другие методы в ранней стадии обнаружения взломов, так как элементарные атаки часто используют заранее известные действия, которые легко распознать. Таким образом, анализ на основе сигнатур быстро и точно информирует о том, что система находится в нормальном состоянии, или вызывает тревогу при необходимости.

Однако анализ на основе сигнатур также имеет свои слабости. Хотя он начинает работать очень быстро, со временем его скорость снижается, поскольку количество обрабатываемых сигнатур увеличивается. Это серьезная проблема, так как число сигнатур может быстро увеличиться с появлением новых атак и действий злоумышленников. Даже использование эффективных методов обработки данных и пакетов не помогает в данном случае, так как некоторые слегка измененные атаки могут пройти через эту систему.

Также есть обратная сторона проблемы: поскольку анализ на основе сигнатур сравнивает пакеты существующими сигнатурами, он способен обнаружить только известные атаки, для которых уже существуют сигнатуры.

Однако стоит отметить, что по статистике 80% атак используют известные сценарии. Наличие сигнатур для известных атак позволяет высокоэффективно обнаруживать вторжения. Поэтому эта технология является достаточно результативной и широко применяется в коммерческих программах.

Преимущества анализа на основе сигнатур:

– обнаружение атак с помощью сигнатур считается достаточно эффективным и при этом не вызывает слишком много ложных срабатываний;

– анализ на основе сигнатур быстро распознает применение конкретной атакующей технологии или инструментов, что помогает администратору улучшить меры безопасности;

– администраторы, независимо от своего уровня квалификации в области безопасности, могут начать процедуры обработки инцидентов с использованием анализа на основе сигнатур;

– системы, которые используют анализ сигнатур известных атак, обладают очень хорошей скоростью работы;

– правила для обнаружения вторжений с использованием сигнатур относительно легко создать, понять и настроить.

Недостатки анализа на основе сигнатур:

– системы, основанные на сигнатурах, должны быть заранее запрограммированы для обнаружения любой атаки и регулярно обновляться с новыми сигнатурами атак;

– самые сигнатуры во многих системах классификации установлены довольно узко, что затрудняет обнаружение вариантов классических атак, которые незначительно отличаются от существующих сигнатур;

– с увеличением числа сигнатур скорость работы системы обнаружения на основе сигнатур будет уменьшаться.

3.4.4. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА ОСНОВЕ ВЫЯВЛЕНИЯ АНОМАЛИЙ

В сети или на сервере нестандартное поведение обычно обнаруживается с помощью детекторов аномалий. Они ищут различия между атаками и «нормальной» активностью, которые можно обнаружить и отследить с помощью системы, способной распознавать эти отклонения.

Детекторы аномалий создают стандартные профили поведения для пользователей, хостов или сетевых подключений на основе исторических данных нормальной работы. Затем они собирают информацию о событиях и анализируют их с помощью различных метрик для определения отклонений от этого стандартного поведения.

Процессы и метрики, используемые при обнаружении отклонений, включают:

– установление допустимого порога в количественных терминах для основных характеристик системы или пользователя. Эти атрибуты могут включать количество файлов, предоставленных пользователю за определенный период времени, использование процессорного времени, неудачные попытки входа в систему и т.д. Уровень порога может быть задан эвристически или статически, в зависимости от изучаемых значений;

– статистические метрики, которые могут быть параметрическими (основанными на заранее определенных шаблонах) или непараметрическими (использующими распределение атрибутов, выведенное из исторических данных за определенный период времени);

– метрики, основанные на правилах, которые определяют возможные образцы атак на основе аномальных данных, но не используют числовые характеристики.

На сегодняшний день только первые две технологии широко применяются в коммерческих системах. Однако детекторы аномалий и основанные на них системы обнаружения угроз часто создают много ложных сигналов из-за неопределенности стандартных действий пользователя или системы. Системы, основанные на аномалиях, способны обнаруживать новые виды атак, в отличие от систем, основанных на сигнатурах, которые полагаются на заранее известные образцы атак.

Кроме того, определенные методы обнаружения аномалий могут предоставлять итоговые данные, которые могут быть использованы в дальнейшем как источники информации для систем обнаружения на основе сигнатур. Например, детектор аномалий на пороге может создавать диаграмму, отображающую «нормальное» количество файлов, предоставленных определенному пользователю, и система обнаружения на основе сигнатур может использовать эту диаграмму как часть своей сигнатуры, чтобы сигнализировать о возможных нарушениях, если количество файлов, доступных пользователю, превышает «нормальную» диаграмму более чем на 10%.

Хотя коммерческие системы обнаружения угроз могут иметь ограниченные возможности обнаружения аномалий, лишь некоторые из них полагаются только на эту технологию. Обнаружение аномалий остается предметом исследования в области активного обнаружения вторжений и, вероятно, станет все более значимым в будущих поколениях систем обнаружения угроз.

Преимущества систем обнаружения на основе аномалий включают:

– возможность обнаруживать неожиданные действия и определять признаки атаки без предварительных знаний об их деталях;

– детекторы аномалий также могут создавать данные, которые могут использоваться для формирования сигнатур для других систем обнаружения.

Недостатки подходов обнаружения аномалий включают:

– необходимость предварительного этапа обучения системы для определения свойств нормального поведения за определенный период времени;

– при данном подходе, как правило, происходит формирование большого количества ложных сигналов при непредсказуемом поведении пользователей и сетевой активности.

3.4.5. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ОСНОВАННЫЕ НА МОДЕЛИРОВАНИИ ПОВЕДЕНИЯ АТАКУЮЩЕГО

Способ организации модели злоупотребления с явными факторами представляет собой один из подходов к выявлению злоупотреблений. Его целью является создание базы данных сценариев атак, которые связывают цепь поведений, характерных для атаки.

При использовании этого подхода осуществляется проверка теории об присутствии одного из сценариев атак в системе путем анализа данных в записях аудита. Итогом поиска считается какое-то количество фактов, достаточное для доказательства или опровержения гипотезы. Обследование производится в одном процессе, получившее название антисипатор.

Антисипатор, работая на основе текущих активных моделей, определяет множество поведений, которые следует исследовать в записях аудита, и передает их планировщику. Планировщик, в свою очередь, определяет, как рассчитанное поведение может воспроизводиться в записях аудита и модифицирует их.

Преимуществом этого метода является возможность сократить число обработок, необходимых для анализа записей аудита. Сначала исследуются наиболее «грубые» действия пассивным образом, а затем, как только хотя бы одно из них обнаружено, анализируются более точные события. Кроме того, планировщик гарантирует независимость представления от формы записей аудита.

Согласно тому, как набираются причины для подозрений определенных сценариев, а для других – снижаются, список моделей активностей уменьшается. Вычисление причин встроено в систему и дает возможность обновлять вероятность возникновения сценариев атак в списке моделей активностей.

Преимущества СОВ, основанных на моделировании поведения атакующего:

– возникает вероятность сократить число значительных обработок, которые нужны только с целью записи аудита; сначала наблюдаются наиболее «грубые» действия в пассивном режиме, и затем, как только лишь одно из них обнаружено, прослеживаются более точные события;

– планировщик гарантирует самостоятельность представления от формы сведений аудита.

Недостатки:

– при использовании описанного подхода у ответственного за формирование модели раскрытия вторжения лица возникает дополнительная нагрузка, связанная с предопределением содержательных

и четких количественных характеристик для различных элементов графического представления модели;

– результативность описанного метода не была показана созданием программного прототипа; из отображения модели не понятно, как действия могут быть эффективно собраны в планировщике, и какой эффект это окажет на систему во время работы;

– данный метод расширяет, но не заменяет подсистему обнаружения аномалий.

3.4.6. СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, ОСНОВАННЫЕ НА АЛЬТЕРНАТИВНЫХ МЕТОДАХ

Новые методы обнаружения вторжений, такие как методы *Data Mining*, технология мобильных агентов, организация иммунных систем, генетические алгоритмы и нейронные сети представляют собой альтернативные подходы к обнаружению атак.

Данные альтернативные подходы возможно представить следующими методами:

– методы *Data Mining*, организованные на основе шаблонов, которые отражают в представленных фрагментах разноаспектных отношений характерные подвыборки сведений, компактно представляющие в удобной человеку форме, закономерности;

– методы технологии мобильных агентов, использующих распределенный сбор необходимых сведений и конкретных функций предварительной обработки от регулируемых хостов с целью получения полного представления о состоянии безопасности в информационных системах. При этом функции исследования и принятия решений поручают единственному механизму;

– методы организации иммунных систем (с помощью аналогии механизмов защиты компьютерных систем и биологическими иммунными системами в отношении способности формировать определение «свой-чужой»);

– методы, основанные на применении генетических алгоритмов;

– методы, которые основаны на использовании нейронных сетей (опирающиеся на искусственный интеллект).

На сегодняшний день они редко используются при проектировании систем вторжений, однако с дальнейшим изучением и развитием этих методов, их активное внедрение в системы обнаружения атак может стать возможным в будущем.

Контрольные вопросы

1. Назовите основные причины проведения удаленных атак на распределенные вычислительные системы.
2. Какими свойствами должна обладать современная СОВ?
3. Какие основные элементы включает в себя современная система обнаружения вторжений?
4. Дайте классификацию систем обнаружений вторжений.
5. Назовите типы систем обнаружения вторжений.

ЗАКЛЮЧЕНИЕ

Данное учебное пособие написано в соответствии с ФГОС специальности 10.05.03 «Информационная безопасность автоматизированных систем» и предназначено для студентов 5 курса, изучающих дисциплину «Программно-аппаратные средства защиты информации». Учебное пособие позволяет студентам облегчить самостоятельное изучение лекционного материала и подготовку к лабораторным работам и экзамену.

Наличие контрольных вопросов по каждой главе учебного пособия позволит студентам самостоятельно проверить свои знания по изученному материалу.

Учебное пособие может быть полезно также студентам 3 курса, обучающимся по направлению подготовки бакалавров 09.03.02 «Информационные системы и технологии» при изучении дисциплины «Основы информационной безопасности» и магистрантов, обучающихся по направлению 09.034.02 «Информационные системы и технологии» при изучении дисциплины «Информационная безопасность и защита информации».

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный государственный образовательный стандарт высшего образования по специальности 10.05.03 «Информационная безопасность автоматизированных систем» (уровень специалитета). Утвержден приказом Министерства образования и науки РФ от 1 декабря 2016 г. № 1509 [электронные данные]. – URL : http://fgosvo.ru/news/1/2131 (дата обращения: 30.01.2017).

2. Баричев, С. Г. Основы современной криптографии / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов. – М. : Горячая линия – Телеком, 2001. – 120 с.

3. ГОСТ 28147–89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования информации. – М. : Госкомитет СССР по стандартам, 1989.

4. ГОСТ Р 34.10–2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронно-цифровой подписи. – М. : Госстандарт России, 2001.

5. ГОСТ Р 34.10–94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма. – М. : Госстандарт России, 1994.

6. ГОСТ Р 34.11–94. Информационная технология. Криптографическая защита информации. Функция хэширования. – М. : Госстандарт России, 1994.

7. ГОСТ Р 51583. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. – М. : Госстандарт России, 1994.

8. Гостехкомиссия России – точка зрения на техническую защиту информации. // JetInfo. – 1999. – № 11. – С. 2 – 12 (www.jetinfo.ru).

9. Грушо, А. А. Теоретические основы защиты информации / А. А. Грушо, Е. Е. Тимонина. – М. : Изд-во Агенства «Яхтсмен», 1996. – 192 с.

10. Программно-аппаратные средства обеспечения информационной безопасности. Теоретические основы компьютерной безопасности :

учебное пособие для вузов / П. Н. Девянин, О. О. Михальский, Д. И. Правиков, А. Ю. Щербаков. – М. : Радио и связь, 2000. – 192 с.

11. Жельников, В. Г. Криптография от папируса до компьютера / В. Г. Жельников. – М. : ABF, 1996. – 336 с.

12. Завгородний, В. И. Комплексная защита информации в компьютерных системах : учебное пособие для вузов / В. И. Завгородний. – М. : Логос, 2001. – 264 с.

13. Зегжда, Д. П. Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. – М. : Горячая Линия – Телеком, 2000. – 452 с.

14. Малюк, А. А. Введение в защиту информации в автоматизированных системах / А. А. Малюк, С. В. Пазизин, Н. С. Погожин. – М. : Горячая Линия – Телеком, 2001. – 148 с.

15. Молдовян, А. А. Криптография / А. А. Молдовян, Н. А. Молдовян, Б. Я. Советов. – СПб. : Лань, 2000.

16. Нечаев, В. И. Элементы криптографии (Основы теории защиты информации) : учебное пособие / В. И. Нечаев ; под ред. В. А. Садовничего. – М. : Высшая школа, 1999. – 109 с.

17. Руководящий документ. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа информации в автоматизированных системах и средствах вычислительной техники // Сборник руководящих документов по защите информации от несанкционированного доступа. – М. : ГТК при президенте РФ, 1998. – 120 с.

18. Руководящий документ. Защита от несанкционированного доступа к информации. Термины и определения // Сборник руководящих документов по защите информации от несанкционированного доступа. – М. : ГТК при президенте РФ, 1998. – 120 с.

19. Саломаа, А. Криптография с открытым ключом / А. Саломаа ; пер. с англ. – М. : Мир, 1995. – 318 с.

20. Мао, В. Современная криптография: теория и практика / В. Мао. – СПб. : Вильямс, 2005.

ОГЛАВЛЕНИЕ

Учебное электронное издание

ГРИДНЕВ Виктор Алексеевич
ГУБСКОВ Юрий Анатольевич
ДЕРЯБИН Андрей Сергеевич
ЯКОВЛЕВ Алексей Вячеславович

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

В ТРЕХ ЧАСТЯХ

ЧАСТЬ 3

Учебное пособие

Редактирование Е. С. Мордасовой
Графический и мультимедийный дизайнер Т. Ю. Зотова
Обложка, упаковка, тиражирование Е. С. Мордасовой

ISBN 978-5-8265-2795-5

9 785826 527955

Подписано к использованию 23.08.2024.
Тираж 50 шт. Заказ № 87

Издательский центр ФГБОУ ВО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Телефон: (4752) 63-81-08
E-mail: izdatelstvo@tstu.ru