Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ № 04 (167)/2007

НОВЫЙ ПОДХОД К ЗАЩИТЕ ИНФОРМАЦИИ – СИСТЕМЫ ОБНАРУЖЕНИЯ КОМПЬЮТЕРНЫХ УГРОЗ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НОВЫЙ ПОДХОД К ЗАЩИТЕ ИНФОРМАЦИИ – СИСТЕМЫ ОБНАРУЖЕНИЯ КОМПЬЮТЕРНЫХ УГРОЗ

доктор техн. наук, профессор А.В. Аграновский,
кандидат техн. наук Р.А. Хади,
г.Ростов-на-Дону, ФГНУ НИИ «Спецвузавтоматика»

СОДЕРЖАНИЕ

Введение

Системы обнаружения сетевых вторжений и выявления признаков компьютерных атак на информационные системы уже давно применяются как один из необходимых рубежей обороны информационных систем. Разработчиками систем защиты информации и консультантами в этой области активно применяются такие понятия (перенесенные из направления обеспечения физической и промышленной безопасности), как защита «по периметру», «стационарная» и «динамическая» защита, стали появляться собственные термины, например, «проактивные» средства защиты.

Исследования в области обнаружения атак на компьютерные сети и системы на самом деле ведутся за рубежом уже больше четверти века. Исследуются признаки атак, разрабатываются и эксплуатируются методы и средства обнаружения попыток несанкционированного проникновения через системы защиты, как межсетевой, так и локальной — на логическом и даже на физическом уровнях. В действительности, сюда можно отнести даже исследования в области ПЭМИН[1], поскольку электромагнитный тамперинг имеет свои прямые аналоги в уже ставшей обычной для рядового компьютерного пользователя сетевой среде. На российском рынке широко представлены коммерческие системы обнаружения вторжений и атак (СОА) иностранных компаний (ISS RealSecure, NetPatrol, Snort, Cisco и т.д.) и в тоже время практически не представлены комплексные решения российских разработчиков. Это вызвано тем, что многие отечественные исследователи и разработчики реализуют СОА, сохраняя аналогии архитектур и типовых решений уже известных систем, не особенно стараясь увеличить эффективность превентивного обнаружения атак и реагирования на них. Конкурентные преимущества в этом сегменте российского рынка достигаются обычно за счет существенного снижения цены и упования на «поддержку отечественного производителя».

На сегодня системы обнаружения вторжений и атак обычно представляют собой программные или аппаратно-программные решения, которые автоматизируют процесс контроля событий, протекающих в компьютерной системе или сети, а также самостоятельно анализируют эти события в поисках признаков проблем безопасности. Поскольку количество различных типов и способов организации несанкционированных проникновений в чужие компьютерные сети за последние годы значительно увеличилось, СОА стали необходимым компонентом инфраструктуры безопасности большинства организаций. Этому способствуют и огромное количество литературы по данному вопросу, которую потенциальные злоумышленники внимательно изучают, и все более изощренные методы и сложные и подходы к обнаружению попыток взлома информационных систем.

Современные системы обнаружения вторжений имеют различную архитектуру. Классификации СОА следует уделить отдельное внимание, поскольку зачастую, используя общепринятую классификацию СОА, специалисты принимают решение о том, какой из программных продуктов применить в той или иной ситуации.

На данный момент можно разделить все системы на сетевые и локальные. Сетевые системы обычно устанавливаются на выделенных для этих целей компьютерах и анализируют трафик, циркулирующий в локальной вычислительной сети. Системные СОА размещаются на отдельных компьютерах, нуждающихся в защите, и анализируют различные события (действия пользователя или программные вызовы). Также различают методики обнаружения аномального поведения и обнаружения злоумышленного поведения пользователей.

Системы обнаружения аномального поведения (от англ. anomaly detection) основаны на том, что СОА известны некоторые признаки, характеризующие правильное или допустимое поведение объекта наблюдения. Под «нормальным» или «правильным» поведением понимаются действия, выполняемые объектом и не противоречащие политике безопасности. Системы обнаружения злоумышленного поведения (misuse detection) основаны на том, что заранее известны некоторые признаки, характеризующие поведение злоумышленника. Наиболее распро-

1 ПЭМИН — побочные электромагнитные излучения и наводки.

страненной реализацией технологии обнаружения злоумышленного поведения являются экспертные системы. Представительным западным аналогом такой системы является бесплатно распространяемая и наиболее популярная система Snort [7].

Классификация компьютерных атак и систем их обнаружения

Эффективная защита от потенциальных сетевых атак невозможна без их детальной классификации, облегчающей их выявление и задачу противодействия им. В настоящее время известно большое количество различных типов классификационных признаков. В качестве таких признаков может быть выбрано, например, разделение на пассивные и активные, внешние и внутренние атаки, умышленные и неумышленные и т.д. [1]. К сожалению, несмотря на то, что некоторые из существующих классификаций мало применимы на практике, их активно используют при выборе СОА и их эксплуатации.

Рассмотрение существующих классификаций начнем с работы Питера Мелла «Компьютерные атаки: что это и как им противостоять» (см. [3]). В ней все возможные сетевые атаки делятся на следующие типы (с детальным описанием атак, приводимых в качестве примеров, можно ознакомиться в книгах «Обнаружение атак», «Атака через Internet», «Атака на Internet», «Атака из Internet» [1, 14-16]):

- удаленное проникновение (от англ. remote penetration) — это тип атак, которые позволяют реализовать удаленное управление компьютером через сеть; например, атаки с использованием программ NetBus или BackOrifice;
- локальное проникновение (от англ. local penetration) — это тип атак, которые приводят к получению несанкционированного доступа к узлу, на который они направлены; примером такой атаки является атака с использованием программы GetAdmin;
- удаленный отказ в обслуживании (от англ. remote denial of service) — тип атак, которые позволяют нарушить функционирование системы в рамках глобальной сети; пример такой атаки — Teardrop или trinOO;
- локальный отказ в обслуживании (от англ. local denial of service) — тип атак, позволяющих нарушить функционирование системы в рамках локальной сети. В качестве примера такой атаки можно привести внедрение и запуск враждебной программы, которая загружает центральный процессор бесконечным циклом, что приводит к невозможности обработки запросов других приложений;
- атаки с использованием сетевых сканеров (от англ. network scanners) — это тип атак, основанных на использовании сетевых сканеров — программ, которые анализируют топологию сети и обнаруживают сервисы, доступные для атаки; пример: атака с использованием утилиты nmap;
- атаки с использованием сканеров уязвимостей (от англ. vulnerability scanners) — тип атак, основанных на использовании сканеров уязвимостей — программ, осуществляющих поиск уязвимостей на узлах сети, которые в дальнейшем могут быть применены для реализации сетевых атак; примерами сетевых сканеров могут служить системы SATAN и Shadow Security Scanner;
- атаки с использованием взломщиков паролей (от англ. password crackers) — это тип атак, которые основаны на использовании взломщиков паролей — программ, подбирающих пароли пользователей; например, программа LOphtCrack для ОС Windows или программа Crack для ОС Unix;
- атаки с использованием анализаторов протоколов (от англ. sniffers) — это тип атак, основанных на использовании анализаторов протоколов — программах, «прослушивающих сетевой трафик. С их помощью можно автоматизировать поиск в сетевом трафике такой информации, как идентификаторы и пароли пользователей, информацию о кредитных картах и т. д. Примерами анализаторов сетевых протоколов являются программы Microsoft Network Monitor, NetXRay компании Network Associates или Lan Explorer.

Приведенная классификация является достаточно полной с практической точки зрения, так как она охватывает почти все возможные

действия злоумышленника. Однако для противодействия сетевым атакам этого недостаточно, так как ее использование в данном виде не позволяет определять элементы сети, подверженные воздействию той или иной атаки, а также последствия, к которым может привести успешная реализация атак. В таком случае мы не включаем в анализ самый важный компонент, а именно — модель угроз безопасности, с построения которой должны начинаться все мероприятия по обеспечению защиты информации.

Аналогичным недостатком страдает и более компактная классификация, предложенная компанией Internet Security Systems, Inc., в которой содержится всего лишь пять типов атак:

- сбор информации (от англ. information gathering);
- попытки несанкционированного доступа (от англ. unauthorized access attempts);
- отказ в обслуживании (от англ. denial of service);
- подозрительная активность (от англ. suspicious activity);
- системные атаки (от англ. system attack).

В своих продуктах, предназначенных для защиты сетей, серверов и рабочих станций (таких как, например, Real Secure, System scanner и др.) компания Internet Security Systems использует несколько других классификационных признаков возможных сетевых атак, они более эффективны с точки зрения защиты от вторжений. Опишем их подробнее.

1. По *степени риска* (от англ. Risk Factor); имеет большое практическое значение, так как позволяет ранжировать опасность атак по следующим классам:
   - высокий (High) — атаки, успешная реализация которых позволяет атакующему немедленно получить доступ к машине, получить права администратора или обойти межсетевые экраны (например, атака, основанная на использовании ошибки в ПО Sendmail версии 8.6.5, позволяет атакующему исполнять любую команду на сервере);
   - средний (Medium) — атаки, успешная реализация которых потенциально может дать атакующему доступ к машине. Например, ошибки в сервере NIS, позволяющие атакующему получить файл с гостевым паролем;
   - низкий (Low) — атаки, при успешной реализации которых атакующий может получить сведения, облегчающие ему задачу взлома данной машины. Например, используя сервис finger, атакующий может определить список пользователей сервера и, используя атаку по словарю, попытаться получить доступ к машине.
2. По *типу атаки* (Attack Type); позволяет судить о том, может ли атака быть осуществлена удаленно, или только локально:
   - осуществляемые локально (Host Based);
   - осуществляемые удаленно (Network Based).
3. По *подверженному данной атаке программному обеспечению* (в англ. варианте Platforms Affected). Например: Microsoft Internet Explorer 5.01, Microsoft Internet Explorer 5.5, Microsoft Internet Explorer 6.

Кроме того, существует классификация *по характеру действий*, используемых в атаке:

- «черные ходы» (Backdoors) — атаки, основанные на использовании недокументированных разработчиками возможностях ПО, которые могут привести к выполнению пользователем несанкционированных операций на атакуемом сервере;
- атаки типа «отказ в обслуживании» (Denial of Service, или DoS) — атаки, основанные на использовании ошибок, позволяющие атакующему сделать какой-либо сервер недоступным для легитимных пользователей;
- распределенные атаки типа «отказ в обслуживании» (Distributed Denial of Service) — несколько пользователей (или программ) посылают большое количество фиктивных запросов на сервер, приводя последний в нерабочее состояние;
- потенциально незащищенная операционная система (OS Sensor);
- неавторизованный доступ (Unauthorized Access Attempts).

К недостаткам приведенных классификационных признаков можно отнести то, что они не позволяют описать цель атаки, а также ее последствия. Например, классификационный признак «по характеру действий» содержит два класса атак типа «отказ в обслуживании», но в то же время не содержит классов, описывающих атак, направленных на перехват трафика.

Другой подход был применен в классификации, использованной в достаточно известном программном продукте Nessus, предназначенном для анализа безопасности серверов. Здесь используется классификация «по характеру уязвимости», используемой для реализации атаки:

- «черные ходы» (Backdoors);
- ошибки в CGI скриптах (CGI abuses);

- атаки типа «отказ в обслуживании» (Denial of Service);
- ошибки в программах – FTP-серверах (FTP);
- наличие на компьютере сервиса Finger или ошибки в программах, реализующих этот сервис (Finger abuses);
- ошибки в реализации межсетевых экранов (Firewalls);
- ошибки, позволяющие пользователю, имеющему терминальный вход на данный сервер, получить права администратора (Gain a shell remotely);
- ошибки, позволяющие атакующему удаленно получить права администратора (Gain root remotely);
- прочие ошибки, не вошедшие в другие категории (Misc);
- ошибки в программах – NIS-серверах (NIS);
- ошибки в программах – RPC-серверах (RPC);
- уязвимости, позволяющие атакующему удаленно получить любой файл с сервера (Remote file access);
- ошибки в программах – SMTP-серверах (SMTP problems);
- неиспользуемые сервисы (Useless services).

Кроме того, по типу программной среды они подразделяются на уязвимости в операционной системе, уязвимости в определенном сервисе и уязвимости в определенном программном обеспечении. Для определения уязвимости в операционной системе используется параметр Host/OS, уязвимости в конкретных сервисах и в определенном программном обеспечении классифицируются по группам.

В этой классификации более детально, по сравнению с предыдущими, проработаны атаки, использующие уязвимости в системном, прикладном и сетевом программном обеспечении. Однако данная классификация не охватывает всех существующих сетевых атак, за пределами рассмотрения остаются такие опасные атаки, как атаки типа «отказ в обслуживании», перехват данных и атаки, направленные на сетевое оборудование. Положительной чертой данной классификации является наличие класса «прочие ошибки, не вошедшие в другие категории», так как формально к любой атаке, в том числе новой, благодаря этому классу будет применима данная классификация. Однако, с другой стороны, этот класс бесполезен, поскольку не содержит никакой дополнительной информации.

Как видно из описанных классификаций, далеко не все они являются полными. В некоторых случаях под видом единой классификации делается попытка объединить несколько классификаций, проведенных по разным характеристическим параметрам.

Появление новых атак приводит к снижению эффективности применения существующих классификаций, поэтому их использование без внесения изменений не представляется возможным. Данная ситуация объясняется огромным количеством различных сетевых атак и постоянным появлением новых атак, некоторые из которых не подчиняются критериям существующих классификаций.

Таким образом, применение существующих классификаций нельзя назвать рациональным. Существует объективная необходимость в создании новой гибкой классификационной схемы возможных сетевых атак, построенной с учетом указанных выше недостатков.

Из отечественных вариантов наиболее информативная и краткая классификация приведена в книге Милославской и Толстого [3]. В ней все СОА делятся на минимальное количество классов – по поведению после обнаружения (на активные и пассивные), по расположению источника результатов аудита (регистрационные файлы хоста либо сетевые пакеты), по методу обнаружения (поведенческие либо интеллектуальные).

Данная классификация наилучшим образом подходит для построения первичных фильтров СОА, поскольку позволяет ответить на вопрос о том, как именно СОА анализируют информацию, как должны различать атаки, какие технологии для этого использовать.

Технологии построения систем обнаружения атак

Системы обнаружения атак, как и большинство современных программных продуктов, должны удовлетворять ряду требований. Это и современные технологии разработки, и ориентиров-

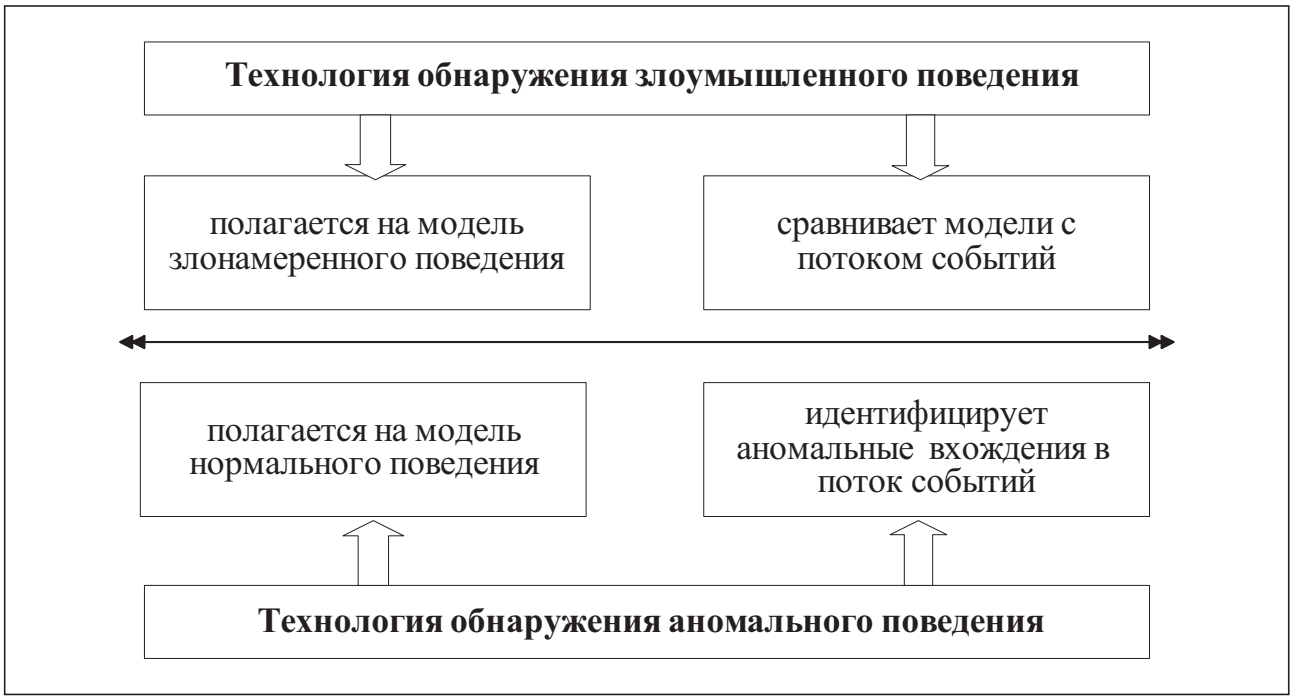

Рис. 1. Существующие технологии СОА

ка на особенности современных информационных сетей, и совместимость с другими программами. Чтобы понять, как правильно использовать СОА, нужно четко представлять, как они работают и каковы их уязвимые места.

Рассмотрим принципы, на которых основана идея обнаружения компьютерных атак. Если не учитывать различные минорные инновации в области обнаружения компьютерных атак, то можно смело утверждать, что существуют две основные технологии построения СОА. Суть их заключается в том, что СОА обладают некоторым набором знаний либо о методах вторжений, либо о «нормальном» поведении наблюдаемого объекта.

Системы обнаружения аномального поведения (anomaly detection) основаны на том, что СОА известны некоторые признаки, характеризующие правильное или допустимое поведение объекта наблюдения. Под нормальным или правильным поведением понимаются действия, выполняемые объектом и не противоречащие политике безопасности.

Системы обнаружения злоумышленного поведения (misuse detection) основаны на том, что СОА известны некоторые признаки, характеризующие поведение злоумышленника. Наиболее распространенной реализацией технологии обнаружения злоумышленного поведения являются экспертные системы (например, системы Snort, RealSecure IDS, Enterasys Advanced Dragon IDS).

Краткая схема, приведенная в статье [9], обобщает эти сведения (см. рис. 1). Все остальные подходы являются подмножествами этих технологий.

Технологии обнаружения аномальной деятельности

Датчики-сенсоры аномалий идентифицируют необычное поведение, аномалии в функционировании отдельного объекта — трудности их применения на практике связаны с нестабильностью самих защищаемых объектов и взаимодействующих с ними внешних объектов. В качестве объекта наблюдения может выступать сеть в целом, отдельный компьютер, сетевая служба (например, файловый сервер FTP), пользователь и т.д. Датчики срабатывают при условии, что нападения отличаются от «обычной» (законной) деятельности. Здесь появляется еще одно слабое место, характерное в большей степени для конкретных реализаций, заключающееся в некорректности определения «дистанции» отклонения наблюдаемого поведения от штатного, принятого в системе, и определении «порога срабатывания» сенсора наблюдения.

Меры и методы, обычно используемые в обнаружении аномалии, включают в себя следующие (согласно [11]):

- пороговые значения: наблюдения за объектом выражаются в виде числовых интервалов. Выход за пределы этих интервалов счи-

тается аномальным поведением. В качестве наблюдаемых параметров могут быть, например, такие: количество файлов, к которым обращается пользователь в данный период времени, число неудачных попыток входа в систему, загрузка центрального процессора и т.п. Пороги могут быть статическими и динамическими (т.е. изменяться, подстраиваясь под конкретную систему);
- статистические меры: решение о наличии атаки делается по большому количеству собранных данных путем их статистической предобработки;
- параметрические: для выявления атак строится специальный «профиль нормальной системы» на основе шаблонов (т.е. некоторой политики, которой обычно должен придерживаться данный объект);
- непараметрические: здесь уже профиль строится на основе наблюдения за объектом в период обучения;
- меры на основе правил (сигнатур): они очень похожи на непараметрические статистические меры. В период обучения составляется представление о нормальном поведении объекта, которое записывается в виде специальных «правил». Получаются сигнатуры «хорошего» поведения объекта;
- другие меры: нейронные сети, генетические алгоритмы, позволяющие классифицировать некоторый набор видимых сенсору-датчику признаков.

В современных СОА в основном используют первые два метода. Следует заметить, что существуют две крайности при использовании данной технологии:
- обнаружение аномального поведения, которое не является атакой, и отнесение его к классу атак (ошибка второго рода);
- пропуск атаки, которая не подпадает под определение аномального поведения (ошибка первого рода). Этот случай гораздо более опасен, чем ложное причисление аномального поведения к классу атак.

Поэтому при инсталляции и эксплуатации систем такой категории обычные пользователи и специалисты сталкиваются с двумя довольно нетривиальными задачами:
- построение профиля объекта — это трудно формализуемая и затратная по времени задача, требующая от специалиста безопасности большой предварительной работы, высокой квалификации и опыта;
- определение граничных значений характеристик поведения субъекта для снижения вероятности появления одного из двух вышеназванных крайних случаев.

Обычно системы обнаружения аномальной активности используют журналы регистрации и текущую деятельность пользователя в качестве источника данных для анализа. Достоинства систем обнаружения атак на основе технологии выявления аномального поведения можно оценить следующим образом:
- системы обнаружения аномалий способны обнаруживать новые типы атак, сигнатуры для которых еще не разработаны;
- они не нуждаются в обновлении сигнатур и правил обнаружения атак;
- обнаружения аномалий генерируют информацию, которая может быть использована в системах обнаружения злоумышленного поведения.

Недостатками систем на основе технологии обнаружения аномального поведения являются следующие:
- системы требуют длительного и качественного обучения;
- системы генерируют много ошибок второго рода;
- системы обычно слишком медленны в работе и требуют большого количества вычислительных ресурсов.

Статистический анализ компьютерных атак

Применение методов статистического анализа является наиболее распространенным видом реализации технологии обнаружения аномального поведения. Статистические датчики собирают различную информацию о типичном поведении объекта и формируют ее в виде профиля. Профиль в данном случае — это набор параметров характеризующих типичное поведение объекта. Существует период начального формирования профиля. Профиль формируется на основе статистики объекта, и для этого могут применяться стандартные методы математической статистики, например метод скользящих окон и метод взвешенных сумм.

После того как профиль сформирован, действия объекта сравниваются с соответствующими параметрами и при обнаружении существенных отклонений подается сигнал о начале атаки. Параметры, которые включаются в про-

филь системы, могут быть отнесены к следующим распространенным группам [12-13]:

- числовые параметры (количество переданных данных по различным протоколам, загрузка центрального процессора, число файлов, к которым осуществлялся доступ и т.п.);
- категориальные параметры (имена файлов, команды пользователя, открытые порты и т.д.);
- не вписывается в классификацию наравне с предыдущими типами параметров.

Профили также должны иметь механизмы динамического изменения, для того чтобы более полно описывать изменяющееся поведение объекта. Системы, применяющие статистические методы, обладают целым рядом достоинств:

- не требуют постоянного обновления базы сигнатур атак. Это значительно облегчает задачу сопровождения данных систем;
- могут обнаруживать неизвестные атаки, сигнатуры для которых еще не написаны. Могут являться своеобразным сдерживающим буфером, пока не будет разработан соответствующий шаблон для экспертных систем;
- позволяют обнаруживать более сложные атаки, чем другие методы. Они могут обнаруживать атаки, распределенные во времени или по объектам нападения;
- могут адаптироваться к изменению поведения пользователя и поэтому являются более чувствительными к попыткам вторжения, чем люди [7].

Среди недостатков систем обнаружения вторжений можно отметить следующие:

- трудность задания порогового значения (выбор этих значений — очень нетривиальная задача, которая требует глубоких знаний контролируемой системы);
- злоумышленник может обмануть систему обнаружения атак, и она воспримет деятельность, соответствующую атаку, в качестве нормальной из-за постепенного изменения режима работы с течением времени и «приручения» системы к новому поведению;
- в статистических методах вероятность получения ложных сообщений об атаке является гораздо более высокой, чем при других методах;
- статистические методы не очень корректно обрабатывают изменения в деятельности пользователя (например, когда менеджер исполняет обязанности подчиненного в критической ситуации). Этот недостаток может представлять большую проблему в организациях, где изменения являются частыми. В результате могут появиться как ложные сообщения об опасности, так и отрицательные ложные сообщения (пропущенные атаки);
- статистические методы не способны обнаружить атаки со стороны субъектов, для которых невозможно описать шаблон типичного поведения;
- системы, построенные исключительно на статистических методах, не справляются с обнаружением атак со стороны субъектов, которые с самого начала выполняют несанкционированные действия. Таким образом, шаблон обычного поведения для них будет включать только атаки;
- статистические методы должны быть предварительно настроены (заданы пороговые значения для каждого параметра, для каждого пользователя);
- статистические методы на основе профиля нечувствительны к порядку следования событий.

Тем не менее, существуют пути решения данных проблем, и их практическая реализация является лишь вопросом времени. Очевидно, что статистический метод является чистой реализацией технологии аномального поведения. Статистический метод наследует у технологии обнаружения аномалий все так необходимые на практике достоинства.

Анализ недостатков современных систем обнаружения вторжений

С учетом сказанного выше, все системы обнаружения вторжений можно разделить на системы, ориентированные на поиск:

- аномалий взаимодействия контролируемых объектов;
- сигнатур всех узнаваемых атак;

- искажения эталонной профильной информации.

Необходимо отметить, что в настоящее время практически отсутствуют системы гибридного типа, а также использующие информацию распределенного во времени и пространстве характера [9]. В ходе работы подавляющего большинства современных систем используется только сигнатурный метод распознавания атакующих воздействий или только поиск аномалий в поведении контролируемой сети.

Еще одним недостатком почти всех известных систем является отсутствие имитатора атак или любого другого средства для проверки корректности развернутой и эксплуатируемой СОА, который обеспечивал бы простое и надежное средство тестирования конфигурационных параметров, использованных в каждой конкретной компьютерной сети.

Данное средство, по логическим соображениям, должно позволять имитировать деятельность программного обеспечения вирусного типа (например, CodeRed, NetSky, Bagle, MSBlast и т.д.), атак на отказ в обслуживании (например, SYN-шторм или атаку типа fraggle), атак с целью повышения привилегий учетной записи (как пример, можно привести уязвимости в сетевых службах MS SQL Server 2000, MS Internet Information Service 5.0), атаку с целью перенаправления трафика и навязывания ложных данных (подмена ARP и навязывание DNS службы). При этом желательно, чтобы программное средство имело возможность генерировать атаки распределенного характера.

Например, архитектура некоторых типов имитаторов СОА состоит из набора агентов различных типов, специализированных для решения подзадач обнаружения вторжений. Агенты размещаются на отдельных компьютерах системы. В данной архитектуре в явном виде отсутствует «центр управления» семейством агентов — в зависимости от сложившейся ситуации ведущим может становиться любой из агентов, инициирующий или реализующий функции кооперации и управления. В случае необходимости агенты могут как клонироваться (осуществлять свое копирование в сетевой и локальной среде), так и прекращать свое функционирование, что очень точно передает характер большинства компьютерных атак. В зависимости от ситуации (вида и количества атак на компьютерные сети, наличия вычислительных ресурсов для выполнения функций защиты), может потребоваться генерация нескольких экземпляров агентов каждого класса. Предполагается, что архитектура системы может адаптироваться к реконфигурации сети, изменению трафика и новым видам атак, используя накопленный опыт.

Архитектура многоагентной системы является интересной и перспективной для дальнейшего рассмотрения. Однако, к сожалению, в отечественных работах нет указаний на используемые или разработанные алгоритмы обнаружения атак. Кроме того, текущие версии известных имитаторов не функционируют в реальном режиме времени (поскольку этого не позволяет делать выбранный базовый инструментарий).

Вообще говоря, отсутствие имитаторов атак для оценки эффективности СОА не является основной проблемой данного направления. Реальными недостатками существующих систем обнаружения компьютерных атак является примитивность простого сигнартурного поиска, малая эффективность при обнаружении распределенных по времени и месту сложных атак, недостаточная интеграция информации на уровне хоста и сети для обнаружения комбинированных атак и несанкционированных проникновений.

Среди эксплуатационных недостатков современных СОА можно отметить большое количество вычислительных операций для простого деления принадлежности события на «свой-чужой» и невозможность обработки всей поступающей информации в реальном режиме времени на обычных персональных компьютерах. Скорость обработки сетевого или иного трафика событий зачастую медленнее реального времени в 1.5-2 раза. А в некоторых системах анализ и вовсе происходит в отложенном режиме. Это означает, что реализация атаки на защищаемые информационные и вычислительные ресурсы не будет замечена вовремя и уж тем более не будет отражена с помощью имеющихся средств защиты. В данном режиме СОА может быть использована в лучшем случае как средство журналирования всех этапов атаки и последующей криминалистической экспертизы.

Большинство современных СОА изначально не разрабатываются «портируемыми», то есть их код непереносим на различные операционные системы и произвольные аппаратно-вычислительные платформы. Работа на нескольких операционных системах для большинства западных продуктов и почти всех отечественных СОА (как экспериментальных, так и коммерчески адаптированных) является невозможной. Учитывая, что СОА не используют преимущества разработки и оптимизации кода для

выбранных операционных систем и аппаратных платформ, это является их одним из самых существенных недостатков.

Кроме того, ни в одной программной или аппаратно-программной системе не предусмотрен режим «горячей замены», позволяющий в случае выведения из строя основного комплекса оперативно ввести в работу комплекс «горячего резервирования» и восстановить уничтоженный рубеж обороны сетевого периметра. Несмотря на это, есть и положительный момент в развитии СОА, который заключается в стремлении разработчиков интегрировать свои системы с существующими средствами защиты (межсетевыми экранами, блокираторами каналов, QoS-диспетчерами).

Анализ систем, использующих сигнатурные методы

Сигнатурные методы позволяют описать атаку набором правил или с помощью формальной модели, в качестве которой может применяться символьная строка, семантическое выражение на специальном языке и т.п. Суть данного метода заключается в использовании специализированной базы данных шаблонов (сигнатур) атак для поиска действий, подпадающих под определение «атака».

Сигнатурный метод может защитить от вирусной или хакерской атаки, когда уже известна сигнатура атаки (например, неизменный фрагмент тела вируса) и она внесена в базу данных СОА. То есть, когда сеть переживает первое нападение извне, первое заражение еще неизвестным вирусом и в базе попросту отсутствует сигнатура для его поиска, сигнатурная СОА не сможет сигнализировать об опасности, поскольку сочтет атакующую деятельность легитимной.

Кроме того, несмотря на кажущуюся простоту сигнатурного метода, и в его реализации есть свои тонкости. Классический пример — с помощью поиска сигнатуры

/../../../../../../../local.ida

и простого сравнения битовой информации невозможно выявить хакерскую атаку на HTTP-сервер. Нападающий может легко изменить строку в соответствии с соглашением об URI и использовать битовую строку

%2F%2E%2F%2E%2F%2E%2F%2E%2F%2E%2F%2Elocal.ida,

которую данная сигнатура уже не охватывает [6].

Большинство существующих программных продуктов, заявляющих об использовании сигнатурного метода, на самом деле реализуют как раз наиболее примитивный способ сигнатурного распознавания. К ним относятся и западные, и практически все отечественные разработки. Многие системы позиционируются как предназначенные для выявления атак в информационных системах на основе интеллектуального анализа сетевых пакетов. На самом же деле сигнатурный метод реализован как алгоритм, исследующий лишь динамику развития атаки, основанный на автомате состояний для оценки сценария развития атаки. По замыслу такой подход должен позволить отследить динамику развития атаки в соответствии с действиями злоумышленника, при этом в качестве модуля сбора данных могут использоваться даже сами системы обнаружения атак.

Однако на практике, например, использование в качестве сбора данных системы Snort сильно замедляет процесс обнаружения, что не позволяет осуществлять анализ в режиме реального времени (хотя оригинальная СОА Snort работает в режиме, близком к реальному времени). С другой стороны, такая система становится очень сложной из-за использования большого количества конфигурационных параметров и переусложнения схемы обработки данных.

Таким образом, эффективность работы сигнатурной СОА определяется тремя основными факторами: оперативностью пополнения сигнутарной базы, ее полнотой с точки зрения определения сигнатур атак, а также наличием интеллектуальных алгоритмов сведения действий атакующих к некоторым базовым шагам, в рамках которых происходит сравнение с сигнатурами.

Для успешной реализации первых двух факторов необходима поддержка международных стандартов и рекомендаций (например, Intrusion Detection Message Exchange Requirements) обмена сигнатурами и информацией об атаках. Поскольку на данный момент не существует достаточно большого количества распределенных и объективных источников сигнатур, то СОА данного типа имеют весьма лимитированную эффективность в реальных сетях.

Анализ систем, использующих методы поиска аномалий в поведении

Системы поиска аномалий идентифицируют необычное поведение («аномалии») в функциони-

ровании контролируемого объекта. В качестве объекта наблюдения может выступать сеть в целом, отдельный компьютер, сетевая служба (например, файловый сервер FTP), пользователь и т.д. Сигнализация СОА срабатывает при условии, что действия, совершаемые при нападении, отличаются от «обычной» (законной) деятельности пользователей и компьютеров. Меры и методы, обычно используемые в обнаружении аномалии, включают использование:

- пороговых значений (наблюдения за объектом выражаются в виде числовых интервалов);
- статистических мер (решение о наличии атаки делается по большому количеству собранных данных);
- профилей (для выявления атак на основе заданной политики безопасности строится специальный список легитимных действий «профиль нормальной системы»);
- нейронных сетей, генетических алгоритмов.

Отличительной чертой данных систем является необходимость их обучения на «стандартное» поведение контролируемого объекта (например, корпоративной интрасети). Это же является и основным недостатком всех подобных методов, поскольку время обучения составляет довольно большой промежуток времени и все это время на контролируемые объекты не должно быть произведено ни единой атаки. В случае, если защищаемая интрасеть на этапе обучения отключается от других сетей, то на этапе эксплуатации система защиты будет классифицировать все попытки легального взаимодействия с внешними сетями как атаки.

В случае создания СОА, использующей профильные системы следует учитывать, что по разным исследованиям, как минимум, 15% пользователей компьютерных сетей не подлежат профилированию вообще, а еще столько же имеют тенденцию к быстрому изменению поведения в течении ограниченного времени. Статичность существующих профильных систем позволяет говорить об этом как об одном из основных недостатков, явно мешающих эксплуатации СОА на базе контроля «профилей» пользователей.

В случае динамической подстройки и модификации профилей необходимо найти компромисс между количеством признаков профилирования (чем их меньше, тем грубее оценивается поведение контролируемого объекта) и скоростью обработки (скорость оценки аномальности поведения по профилю является экспоненциальной функцией от количества исследуемых признаков). Кроме того, большое число конфигурационных параметров в этом случае неизбежно потребуют от администратора системы защиты высокой квалификации в весьма специализированной области обнаружения атак.

Такой подход реализован в некоторых отечественных СОА. Данные разработки относятся к классу системных СОА, их экземпляры должны эксплуатироваться на каждом информационном ресурсе, нуждающемся в защите. Особенностью одной из данных систем является использование процедур нечеткого поиска. Для каждого из пользователей создается свой индивидуальный профиль, при этом поведение, характерное для одного из пользователей, может считаться необычным для другого, и наоборот. Поскольку такие профили трудно формализовать, они создаются на основе примеров нормальной работы того или иного пользователя. В качестве показателей активности пользователей выбраны запуск и завершение приложений, а также переход от одного активного приложения к другому. Профили создаются на основе примеров нормальной работы того или иного пользователя. Для представления профилей разработчиками были выбраны нейронные сети. По данным разработчиков, тестирование системы показало, что вероятность ошибки первого рода составляет 5-15%, ошибки второго рода — 10-20%. При этом до половины тестовой выборки составляли вектора пользователей, которые не участвовали в построении обучающей выборки, что говорит о хорошей обучающей способности нейронной сети.

Общая оценка современного подхода к обнаружению вторжений

Большинство рассмотренных недостатков современных СОА являются недостатками, с которыми может столкнуться пользователь в реальных компьютерных сетях. Большая часть замечаний о недостатках и степени эффективности разрабатываемых методов и средств происходит из практики использования СОА в реальных корпоративных интрасетях.

Существующие подходы к решению задач обнаружения вторжений зачастую отличаются не только реализацией методов обнаружения, но и своей архитектурой, уровнем детализации и типами обнаружения вторжений. Естественно, что у каждой системы есть свои достоинства и недостатки. Несмотря на постоянное развитие применяемых при разработке СОА технологий, о легкости развертывания, эксплуатации и мо-

дификации систем обнаружения вторжений придется забыть, все существующие разработки имеют тенденцию лишь к усложнению. Технологии взлома постоянно совершенствуются, атаки становятся комбинированными и распространяются с очень большой скоростью, поэтому к современным СОА выдвигаются все более жесткие и сильные требования. Очевидно, что для соответствия своей задаче СОА должны реализовывать две основные рассмотренные выше технологии, в той или иной степени взаимодополняющие друг друга.

Если рассматривать СОА с точки зрения методов обнаружения атак, то, очевидно, это должны быть системы, включающие в себя множество модулей, реализующих различные подходы — с учетом различных типовых сегментов защищаемых сетей. Перед большинством СОА уже стоит проблема повышения быстродействия, так как современные компьютерные сети становятся все более быстрыми. По мере внедрения СОА в эксплуатацию повышаются требования к масштабируемости и простоте управления системами обнаружения. В будущем СОА, видимо, разделятся на две категории, которые будут использовать различные подходы для малых корпоративных сетей и для больших, сложных по своей топологии компьютерных интрасетей территориально распределенных корпораций.

Таким образом, требования и особенности современных компьютерных сетей, такие как повышение надежности сетей, повышение мобильности, иерархическая структура сетей, различные требования к безопасности — все это накладывает отпечаток на технологии и подходы, которые должны быть уже сегодня реализованы в системах обнаружения атак.

Концепция обнаружения компьютерных угроз, а не атак

При построении современной системы обнаружения вторжений необходимо, прежде всего, сформировать правильные взгляды на информационные процессы, проходящие не только в компьютерной сети, но и во всей информационной системе (ИС). Система обнаружения компьютерных вторжений и атак, по сути, является специализированной системой обработки информации, предназначенной для чрезвычайно быстрого анализа огромного объема данных совершенно разного вида. Для того чтобы определить наиболее точно критерии эффективности такой системы и оценить параметры, которые наиболее сильно влияют на скорость и точность работы, необходимо проанализировать — какого рода данные будут обрабатываться в системе и каким образом это должно происходить.

При этом следует учитывать тот факт, что система обнаружения атак должна функционировать адекватно угрозам безопасности, характерным для рассматриваемых объектов информационной системы, поэтому исходной позицией является выявление перечня угроз, характерных для данной ИС.

К сожалению, практически все существующие системы обнаружения компьютерных атак лишены функциональности, позволяющей связывать риски и угрозы безопасности с происходящими в сетевой и локальной вычислительной среде событиями. В результате такого одностороннего анализа, когда в расчет принимаются только технические параметры сети, причем их весьма ограниченное количество, страдает в первую очередь качество обнаружения атак.

Более того, пользователь такой системы никогда не получит той информации, ради которой эти системы эксплуатируются — информации о реализации угроз безопасности, которым подвержена защищаемая сетевая и локальная инфраструктура.

Обнаружение угроз безопасности

Для описания нового подхода введем понятия, которые будут применяться в дальнейшем. Под информационной системой в данной работе будет пониматься совокупность технических средств (компьютеров, коммуникационного оборудования, линий передачи данных), при помощи которых обеспечивается обработка информации в организации.

Под угрозой информационной системе будем понимать потенциально возможное действие, предпринимаемое злоумышленником, которое может привести к прямому или косвенному ущербу.

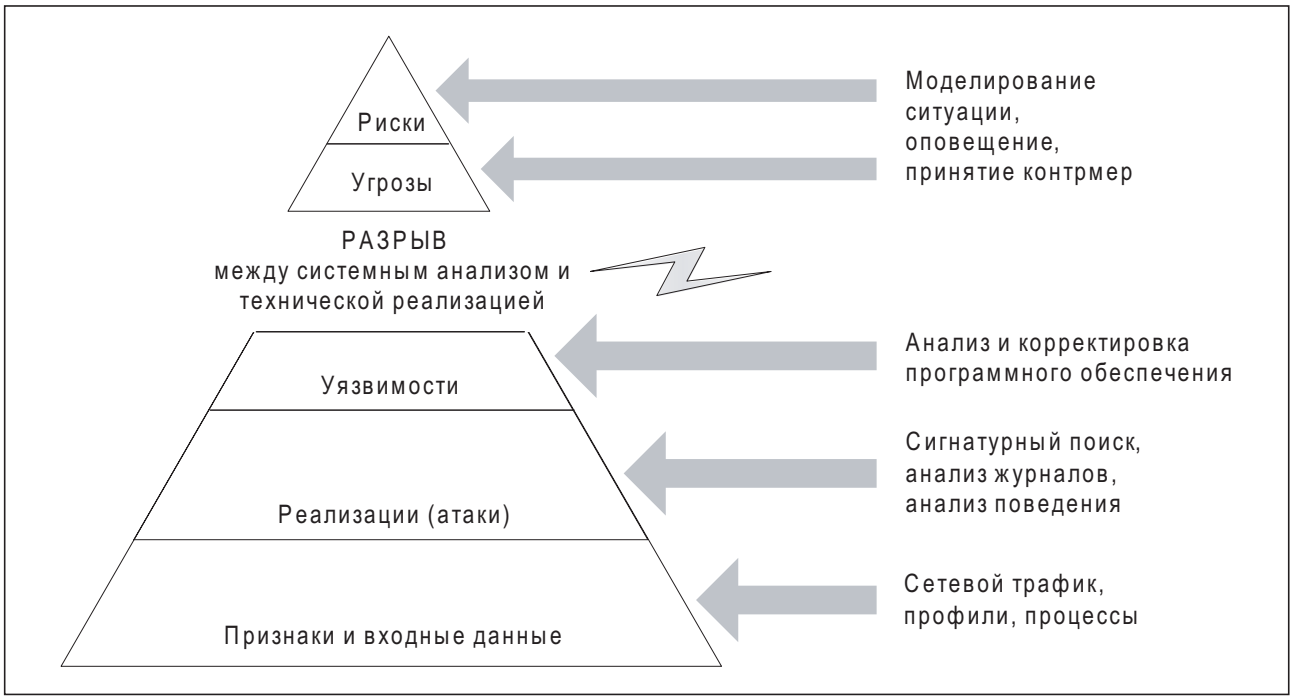

Рис. 2. Информационная пирамида

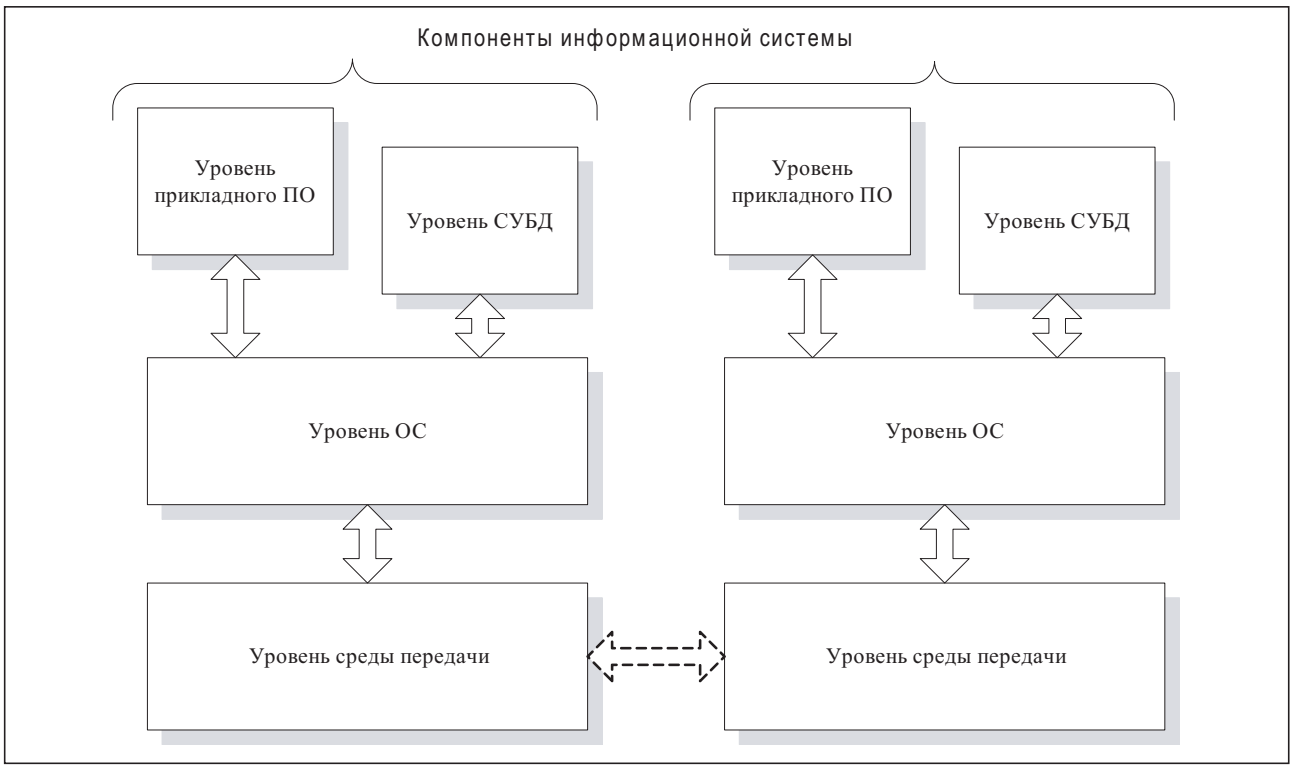

Рис. 3. Уровни обработки информации в информационной системе

Атакой на ИС будем называть действие или некоторую последовательность действий, предпринимаемых злоумышленником для достижения несанкционированного результата, в обход установленных политик безопасности. В нашем предложении рассматриваются действия, направленные на нарушение установленных владельцем правил функционирования системы, выполняемые при помощи различных средств вычислительной техники.

Целью предлагаемой концепции обнаружения угроз информационной безопасности

является определение новых требований и принципов конструирования систем обнаружения компьютерных атак, ориентированных на комплексную обработку информации о защищаемой инфраструктуре для своевременного выявления и предупреждения о возможности реализации угроз, присущих информационной системе.

На сегодня пирамида информационной обработки данных в современной СОА выглядит следующим образом (см. рис. 2).

Верхняя часть информационной пирамиды — это риски и угрозы, присущие рассматриваемой системе. Ниже располагаются различные варианты реализаций угроз (атаки), и самый нижний уровень — это признаки атак. Конечный пользователь, равно как и система обнаружения атак, имеет возможность регистрировать только процесс развития конкретной атаки или свершившийся факт атаки по наблюдаемым характерным признакам. Признаки атаки — то, что мы реально можем зафиксировать и обработать различными техническими средствами, а следовательно, необходимы средства фиксации признаков атак.

Если данный процесс рассматривать во времени, то можно говорить, что определенные последовательности наблюдаемых признаков порождают события безопасности. События безопасности могут переводить защищаемые объекты информационной системы в небезопасное состояние. Следовательно, для системы обнаружения атак необходим информационный срез достаточной полноты, содержащий все события безопасности, произошедшие в информационной системе за рассматриваемый период. Кроме того, поднимаясь вверх по пирамиде, для события безопасности можно указать, к реализации какого вида угроз оно может привести, для того чтобы в процессе развития атаки производить прогнозирование ее развития и принимать меры по противодействию угрозам, которые может вызывать данная атака.

Методология обработки данных в современных информационных системах подразумевает повсеместное использование многоуровневости. Для СОА нового типа можно выделить следующие крупные уровни, на которых возможно осуществление доступа к обрабатываемой информации:

1. Уровень прикладного ПО, с которым работает конечный пользователь информационной системы. Прикладное программное обеспечение зачастую имеет уязвимости, которые могут использовать злоумышленники для доступа к обрабатываемым данным ПО.
2. Уровень СУБД. Уровень СУБД является частным случаем средств прикладного уровня, но должен выделяться в отдельный класс в силу своей специфики. СУБД, как правило, имеет свою собственную систему политик безопасности и организации доступа пользователей, которую нельзя не учитывать при организации защиты.
3. Уровень операционной системы. Операционная система компьютеров защищаемой ИС является важным звеном защиты, поскольку любое прикладное ПО использует средства, предоставляемые именно ОС. Бесполезно совершенствовать качество и надежность прикладного ПО, если оно эксплуатируется на незащищенной ОС.
4. Уровень среды передачи. Современные ИС подразумевают использование различных сред передачи данных для взаимосвязи аппаратных компонентов, входящих в состав ИС. Среды передачи данных являются на сегодня одними из самых незащищенных компонентов ИС. Контроль среды передачи и передаваемых данных является одной из обязательных составляющих механизмов защиты данных.

Иллюстративно уровни обработки потоков данных в информационной системе представлены на рис. 3.

Исходя из вышесказанного, можно сделать вывод, что любые средства защиты информации, в том числе и системы обнаружения и предупреждения атак, обязаны иметь возможность анализировать обрабатываемые и передаваемые данные на каждом из выделенных уровней. Требование присутствия системы обнаружения атак на каждом функциональном уровне информационной системы приводит к необходимости выделения подсистемы регистрации событий безопасности в отдельный комплекс информационных зондов СОА, обеспечивающих сбор информации в рамках всей сети информационной системы. В то же время, разнородность программно-аппаратных платформ и задач, решаемых различными объектами ИС, требует применения модульной архитектуры информационных зондов для обеспечения возможности максимальной адаптации к конкретным условиям применения.

Использование знаний об угрозах ИБ для обнаружения атак на информационную систему

Угрозы информационной безопасности, как правило, каким-либо образом взаимосвязаны друг с другом. Например, угроза захвата уязвимого веб-сервера узла сети может привести к реализации угрозы полного захвата управления данным узлом, поэтому в целях прогнозирования и оценки ситуации целесообразно учитывать вероятностную взаимосвязь угроз.

Если рассмотреть U — множество угроз безопасности рассматриваемой информационной системы, то $u_i \in U$ — i-я угроза. В предположении, что множество угроз конечно, будем считать, что реализация i-ой угрозы может с некоторой вероятностью приводить к возможности реализации других угроз. При этом возникает задача вычисления $P(u|u_{i1},u_{i2},...,u_{ik})$ — вероятности реализации угрозы u, при условии реализации угроз $u_{i1},u_{i2},...,u_{ik}$ (см. рис. 4).

Наиболее надежно атаку можно обнаружить, имея как можно более полную информацию о произошедшем событии. Как видно из предыдущих разделов, современные системы чаще всего фиксируют атаки по наличию определенной, вполне конкретной сигнатуры.

Расширив этот подход, мы можем акцентировать внимание на процесс выделения в компьютерных атаках различных этапов (фаз) их реализации [1]. Выделение фаз атак, особенно ранних, является важным процессом, который, в конечном счете, позволяет обнаружить атаку в процессе ее развития. Однако сделать это возможно лишь определив соответствующим образом перечень угроз информационной системе, которые могут реализовываться на каждой из фаз атаки, и соответствующим образом отразив данный факт в классификации. В самом крупном приближении выделяются три основных фазы атаки: сетевая разведка, реализация, закрепление и сокрытие следов.

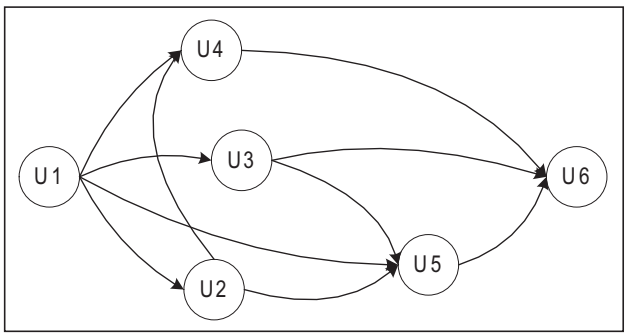

Рис. 4. Вид графа зависимости угроз ИБ

Анализ взаимосвязи угроз с фазами атаки и прогнозирования наиболее вероятных угроз, которые могут быть реализованы злоумышленником, является важной задачей обеспечения ИБ. Это необходимо для своевременного принятия решений по блокировке злонамеренных воздействий.

Следующим элементом концепции обнаружения атак является классификация. Вопросы классификации компьютерных атак до сих пор активно исследуются. Основная задача разработки классификации компьютерных атак состоит в том, чтобы обеспечить удобство использования данной классификации на практике. Основные требования к классификации таковы: непересекающиеся классы, полнота, применимость, объективность, расширяемость, конечность. Интересные подходы к классификации сетевых атак предложены в [2]. Классификация угроз безопасности должна учитывать структуру и фазы проведения атаки на компьютерные системы, определять такие атрибуты как источники и цели атаки, их дополнительные характеристики, многоуровневую типизацию. Модель обнаружения вторжений должна строиться на базе разработанной классификации.

Таким образом, в перспективе необходимо решение следующих задач — определение наиболее вероятной реализации угрозы на текущий момент времени для того, чтобы иметь представление, какие последствия могут в кратчайшее время ожидать информационную систему, а также составление прогноза развития ситуации с целью определения наиболее вероятной реализации угроз в будущем.

Повышение эффективности систем обнаружения атак – интегральный подход

Вообще говоря, современные системы обнаружения вторжений и атак еще далеки от эргономичных и эффективных, с точки зрения безопасности решений. Повышение же эффективности следует ввести не только в области обнаружения злонамеренных воздействий на инфраструктуру защищаемых объектов информатизации, но и с точки зрения повседневной «боевой» эксплуатации данных средств, а также экономии вычислительных и информационных ресурсов владельца данной системы защиты.

Если же говорить непосредственно о модулях обработки данных, то, следуя логике предыдущего раздела, каждая сигнатура атаки в представленной схеме обработки информации об атаке является базовым элементом для распознавания более общих действий — распознавания фазы атаки (этапа ее реализации). Само понятие сигнатуры обобщается до некоторого решающего правила (например, с помощью поиска аномалий в сетевом трафике или клавиатурном почерке пользователя). А каждая атака наоборот разбивается на набор этапов ее проведения. Чем проще атака, тем проще ее обнаружить и больше возможностей появляется по ее анализу. Каждая сигнатура отображает определенное событие в вычислительной сетевой и локальной среде в фазовое пространство компьютерных атак. Фазы можно определить свободно, но лучше сохранять при этом достаточную степень детализации, чтобы иметь возможность описывать атаки с помощью подробных сценариев атак (списка фаз атак и переходов между ними).

Сценарий атаки в этом случае представляет собой граф переходов, в аналогичный графу конечного детерминированного автомата. А фазы атак можно описать, например, следующим образом:

- опробование портов;
- идентификация программных и аппаратных средств;
- сбор баннеров;
- применение эксплоитов;
- дезорганизация функционала сети с помощью атак на отказ в обслуживании;
- управление через бэкдоры;
- поиск установленных троянов;
- поиск прокси-серверов;
- удаление следов присутствия;
- и т.д. (по необходимости с различной степенью детализации).

Преимущества такого подхода очевидны — в случае раздельной обработки различных этапов атаки появляется возможность распознавать угрозу еще в процессе ее подготовки и формирования, а не на стадии ее реализации, как это происходит в существующих системах. При этом, элементной базой для распознавания может быть как сигнатурный поиск, так и выявление аномалий, использование экспертных методов и систем, доверительных отношений и прочих информационных, уже известных и реализованных, сетевых и локальных примитивов оценки происходящего в вычислительной среде потока событий.

Обобщающий подход к анализу позволяет соответственно определять и распределенные (во всех смыслах) угрозы, как во временном, так и логическом и физическом пространстве. Общая схема обработки поступающих событий также позволяет осуществлять поиск распределенных атак — путем последующей агрегации данных из различных источников и конструирования мета-данных об известных инцидентах по защищаемому «периметру» (см. рис. 5).

Распределенные атаки выявляются путем агрегации данных о поступающих атаках и подозрительных действиях и сопоставления шаблонов и статистической фильтрации. Таким образом, оповещение о подозрительных действиях в компьютерных системах происходит на нескольких уровнях:

- нижний уровень сообщает о примитивных событиях (совпадении сигнатур, выявлении аномалий);
- средний уровень извлекает информацию из нижнего уровня и агрегирует ее с помощью конечных автоматов (сценариев атак), статистического анализа и механизмов пороговой фильтрации;
- высший уровень агрегирует информацию с двух предыдущих и позволяет выявлять обычные и распределенные атаки, их реальный источник и прогнозировать его дальнейшее поведение на основе интеллектуального анализа.

Ядро системы обнаружения компьютерных атак должно быть четко разделено с системой визуализации и сигнализации.

Для поиска сигнатур в сетевых пакетах используются правила, формирующие перечень опций (паспорт), по которым осуществляется проверка поступающих сетевых пакетов. Существующие системы (как, например, Snort или PreludeIDS, которая использует правила Snort) применяют строчный вид описаний таких правил:

```
alert tcp $HOME_NET 1024:65535 ->
$EXTERNAL_NET 1024:65535 (msg:"BLEEDING-
EDGE TROJAN Trojan.Win32.Qhost C&C Traffic
Outbound (case1)"; flow:established; dsize:>1000;
content:"|00 00 00 28 0a 00 00 02 0f|Service Pack
1|00|"; classtype:trojan-activity; reference:
url,/www.viruslist.com/en/viruses/
encyclopedia?virusid=142254; sid:2007578; rev:1;)
```

Такой вид более удобен для быстрой машинной обработки, но менее пригоден для человека. Кроме того, в нем отсутствуют возможно-

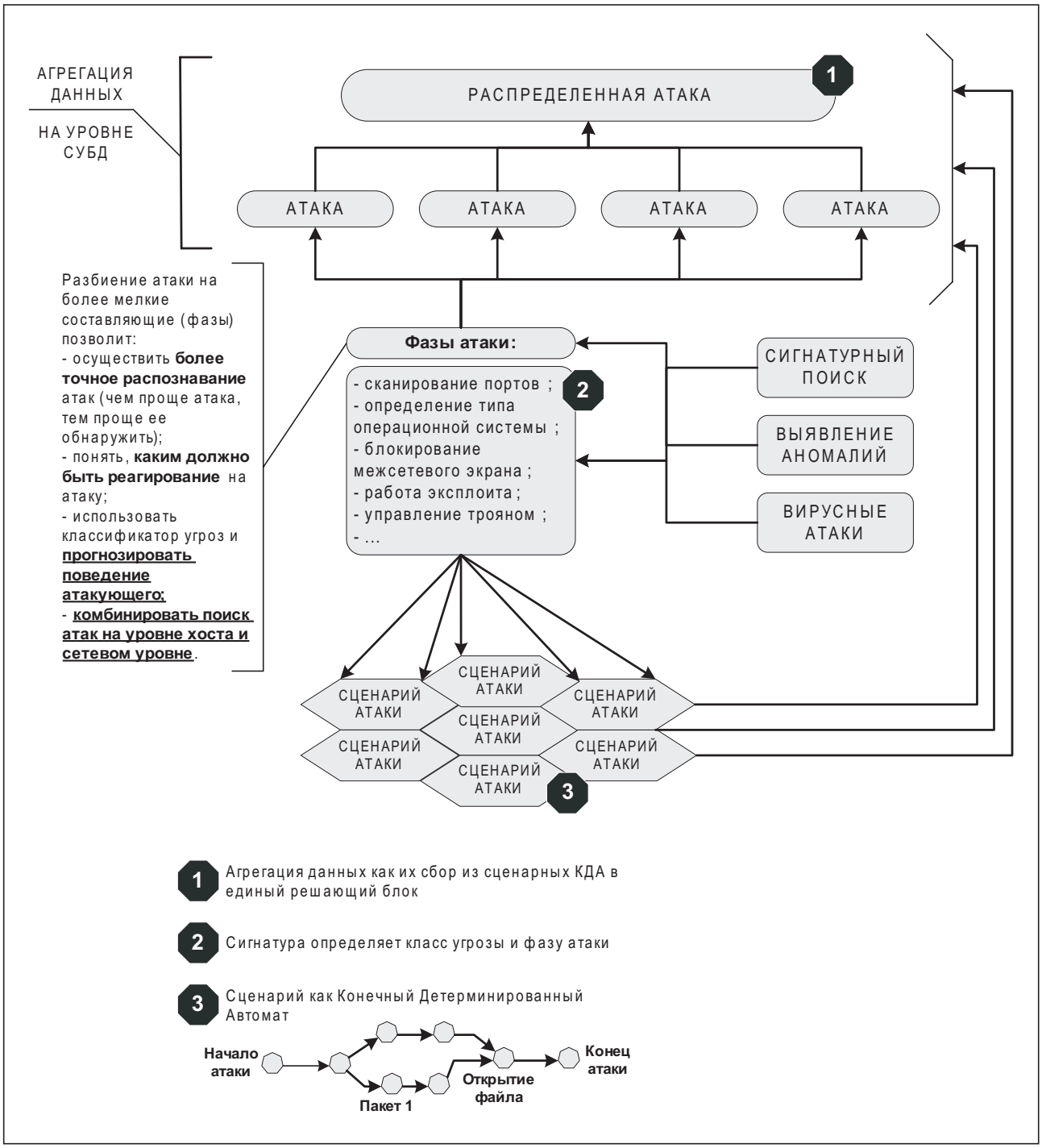

Рис. 5. Схема интегрального обнаружения компьютерных атак

сти для расширения функциональности, которые заложены в XML-подобных реализациях сигнатурных баз. Например, простая «скобочная» (от англ. brace-like) конфигурация позволяет записать ряд управляющих переменных и описать правила в гораздо более приятной и понятной визуальной форме, сохраняя возможность для легкого расширения функциональности. Так, определение фаз атак, защищаемых объектов и совершаемых в сети событий может выглядеть следующим образом:

```
type_defs {
      alert = 1;
      warning = 2;
      fail = 4;
}
srcdst_defs {
```

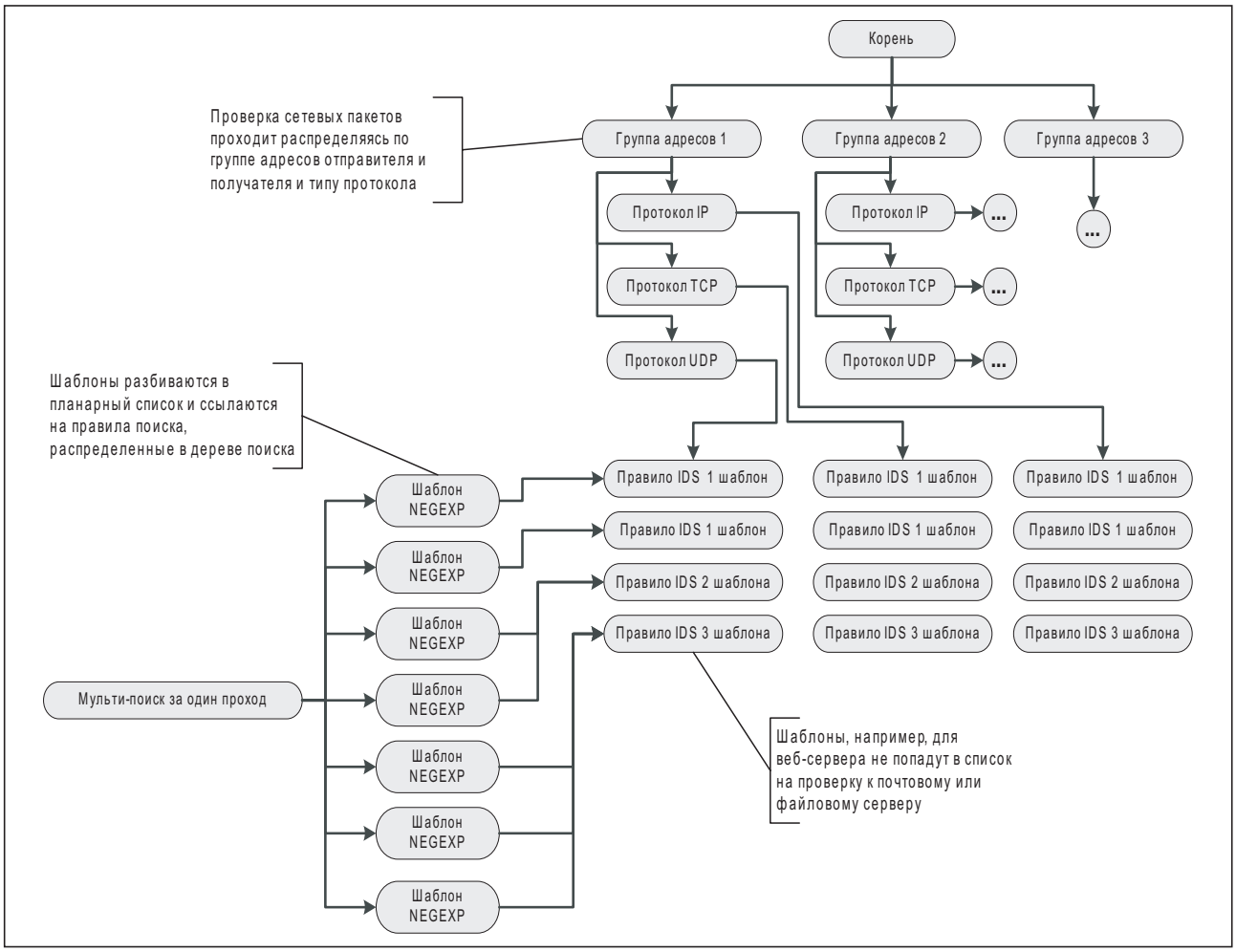

Рис. 6. Схема интегрального обнаружения компьютерных атак

```
      HOME_NET = 195.208.245.212
localhost = 127.0.0.1
}
proto_defs {
      tcp = 1;
      udp = 2;
      tcp-flow = 10;
}
phase_defs {
      port_scanning = 1;
      exploiting = 2;
      icmp_sweeping = 3;
      ftp_bouncing = 4;
      shell_using = 5;
      dir_listing = 6;
      file_opening = 7;
}
```

А секция определения угроз информационной безопасности может иметь основные позиции, подобные следующей:

```
treat_defs = {
     treat {
        name = file-unauthorised-access;
        id = FUAC;
        msg = «message in english»;
      }
  }
```

Кроме указанных в гибкой форме угроз, фаз атак и защищаемых объектов, интегральная обработка информации, связанная с выявлением угроз информационной безопасности, позволяет ввести также сервис-ориентированный подход к обнаружению атак, формируя автоматическим или ручным способом описания сетевых и локальных служб, а также приоритезируя важность, с точки зрения обеспечения должного уровня, информационной безопасности и жизнедеятельности информационной инфраструктуры сети.

```
service_defs = {
   service {
      name = pop3;
      msg = «»;
      rulesets = «backdoors, pop3scanners»;
```

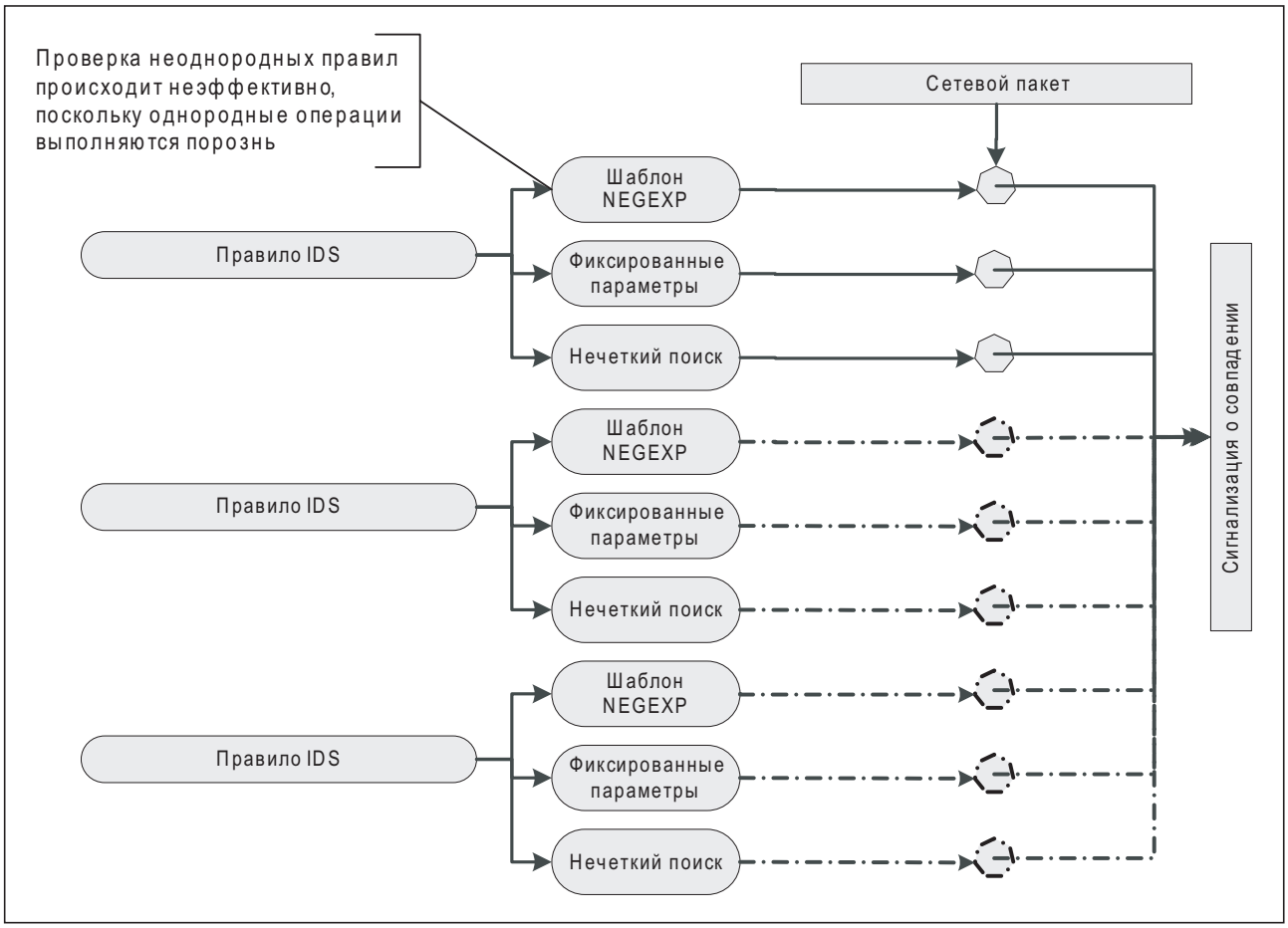

Рис. 7. Неэффективный способ проверки

```
      security_tolerance = 3
      life_insurance = 5
   }
}
```

Сами же правила выглядят, например, следующим образом:

```
ruleset {
   name = backdoors;
   rule {
      id = 0x1000;
      type = alert;
      proto = tcp;
      src = localhost;
      dst = 195.208.245.0/24:2000;
      msg = «service::what is bad in this alert»;
      options = AP,vice_versa;
      contains = «|0a0a0d03|»;
      phase = exploiting;
      treat = file-unauthorized-access;
      revision = 1;
   }
}
```

Здесь учитываются как классические признаки события (тип события, протокол обнаружения, источник и объект воздействия, краткое сообщение), так и добавочные — фаза атаки, тип угрозы, к возникновению которой относится данное событие. При этом сами правила могут быть сгруппированы в наборы, пригодные затем для связывания их с установленными в защищаемой системе сетевыми и локальными службами.

Если же вернуться к эффективности проверки правил в системах обнаружения сетевых атак, то следует отметить следующий факт. На текущий момент все правила в системах СОА проверяются следующим образом (см. рис. 7). Проверка неоднородных правил происходит раздельно, правило за правилом, при этом однородные операции над пакетами выполняются все время порознь. Такой подход не позволяет эффективно распараллелить обработку сетевых пакетов, полностью использовать возможности нескольких конвейеров на современных процессорах, а также оптимизировать поиск частично похожих правил-сигнатур.

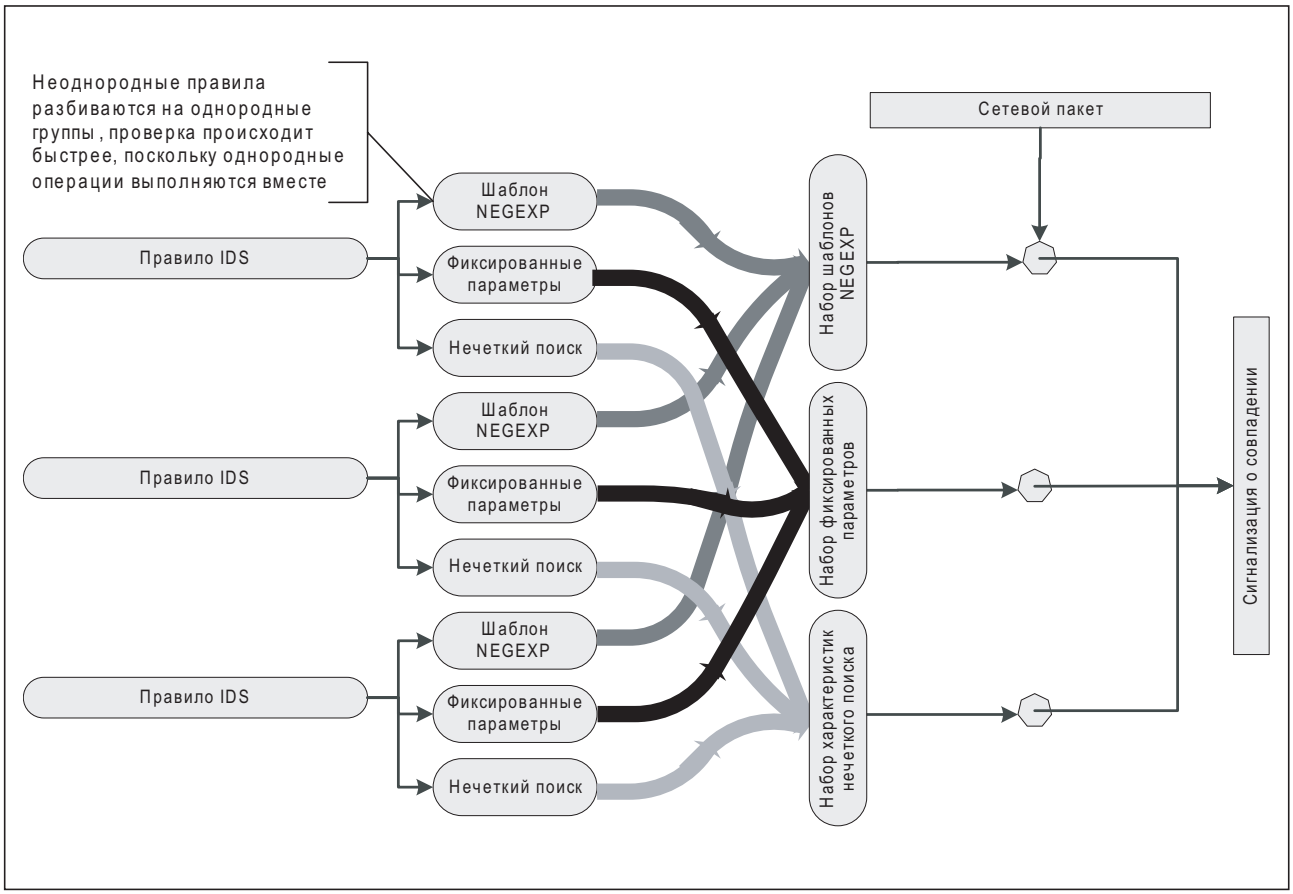

Рис. 8. Метод послойной проверки

Для того чтобы ликвидировать эти недостатки, предлагается однородные операции сводить в единые списки со ссылками на исходные правила (см. рис. 8). Параллельность обработки в этом случае достигается за счет использования «корзин» и «штрафных мячей» (каждое правило имеет пустую корзину, а за каждое совпадение шаблона или другой части правила в корзину добавляется один мяч; при достижении определенного количества мячей правило считается совпавшим). Единство списков позволяет убрать неэффективные, совпадающие в разных правилах, шаблоны.

Однако есть минус и такого подхода, когда, например, шаблоны связаны друг с другом (вот пример такого шаблона: найти первое вхождение, затем относительно него через несколько байт проверить наличие определенной бинарной последовательности). Правда, таких правил — подавляющее меньшинство (даже если судить по общепринятым правилам популярной СОА Snort), что позволяет вынести их в отдельный класс распараллеливаемых методов и использовать в них любые простые методы последовательной проверки.

Помимо преимущества в распараллеливании процесса поиска сигнатур, становится возможным применение методов одновременного поиска многих сигнатур в сетевом потоке за один проход (можно, например, построить один большой конечный автомат для большинства шаблонов, участвующих в правилах, или использовать мультисигнатурную модернизацию алгоритма Бойера-Мура).

Экспериментальные проверки различных вариантов реализации методов одновременного поиска многих сигнатур показали, что наиболее быстрой оказывается реализация большого конечного автомата, модифицированного таким образом, чтобы он позволял «пропускать» однородные ошибки — пропуски и вставки произвольной длины, а также ошибки замены (в результате модификации сигнатуры, что является довольно частым явлением, с целью ее сокрытия от СОА).

Наиболее сложные в проверке правила (шаблоны) можно предварительно компилировать в бинарные подключаемые модули (как это сделано, например, в системе RealSecure IDS).

Заключение

Современный подход к построению систем обнаружения сетевых вторжений и выявления признаков компьютерных атак на информационные системы полон недостатков и уязвимостей, позволяющих, к сожалению, злонамеренным воздействиям успешно преодолевать системы защиты информации. Переход от поиска сигнатур атак к выявлению предпосылок возникновения угроз информационной безопасности должен способствовать тому, чтобы в корне изменить данную ситуацию, сократив дистанцию отставания в развитии систем защиты от систем их преодоления.

Кроме того, такой переход должен способствовать повышению эффективности управления информационной безопасностью и, наконец, более конкретным примерам применения нормативных и руководящих документов уже ставших стандартами.

Литература

1. Лукацкий А.В. Обнаружение атак — СПб.: БХВ-Петербург, 2001. — 624 с.: ил.
2. Климовский А.А. К анализу подходов классификации компьютерных атак // Материалы международной научной конференции по проблемам безопасности и противодействия терроризму. — М.:МЦНМО, 2006 — 480с.
3. Милославская Н.Г., Толстой А.И. Интрасети: обнаружение вторжений. Учебное пособие для вузов. — М.: ЮНИТИ-ДАНА, 2001. — 587с.
4. Сердюк В.А. Анализ современных тенденций построения моделей информационных атак // Информационные технологии, №4, 2004.
5. Новиков А.А., Устинов Г.Р. Уязвимость и информационная безопасность телекоммуникационных технологий. М: Радио и связь, 2003 296с.
6. Huang M., Wicks T.M. A Large-scale Distributed Intrusion Detection Framework Based on Attack Strategy Analysis // In Proceedings of First International Workshop on the Recent Advances in Intrusion Detection, September 14-16, 1998, Louvain-la-Neuve, Belgium.
7. Чирилло Дж. Обнаружение хакерских атак. СПб.: Питер, 2002. — 864с.
8. Eiter M. V. Comparing environments for developing agents. Technical report, Technische Universitat Wien, March 2001.
9. Аграновский А.В., Хади Р.А., Балакин А.В. Обучаемые системы обнаружения и защиты от вторжений // Искусственный интеллект, N3, Донецк, Украина, 2001, стр. 440-444.
10. Bace R., Mell P. Intrusion Detection Systems // NIST Special Publication on Intrusion Detection Systems, 2001.
11. Teresa L.F. A Survey of Intrusion Detection Techniques // Computers and Security 12, 4 (June 1993): 405-418.
12. Kotenko I., Man'kov E. Agent-Based Modeling and Simulation of Computer Network Attacks // Proceedings of Fourth International Workshop Agent-Based Simulation 4 (ABS 4), Montpellier, France, 2003, p.121-126.
13. Kotenko I. Teamwork of Hackers-Agents: Modeling and Simulation of Coordinated Distributed Attacks on Computer Networks // Proceedings of the 3rd International/Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003). Prague, Czech Republic, 2003, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol.2691, p.464-474.
14. Медведовский И.Д., Семьянов П.В., Платонов В.В. Атака через Internet. — М.: НПО «Мир и семья-95, 1997. — 296с.
15. Медведовский И.Д., Семьянов П.В., Леонов Д.Г. Атака на Internet. — М.:ДМК, 1999. — 336с.
16. Медведовский И.Д., Семьянов П.В., Леонов Д.Г., Лукацкий А.В. Атака из Internet.- М.:Солон-Р, 2002. — 368с.

Jet Info
ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ
Издается с 1995 года
Издатель: компания «Инфосистемы Джет»

Главный редактор: Дмитриев В.Ю. *(vlad@jet.msk.su)*
Редактор: Лапина И.К. *(lapina@jet.msk.su)*
Россия, 127015, Москва, Б. Новодмитровская, 14/1
тел. (495) 411 76 01
факс (495) 411 76 02
email: JetInfo@jet.msk.su http://www.jetinfo.ru

Подписной индекс по каталогу Роспечати 32555