Лекция 16
Комплекс программно-технических средств для обеспечения ИБ

В состав программно-технических средств, совместное применение которых обеспечивает комплексную защиту КСС входят:

- средства построения VPN,
- межсетевые экраны,
- средства криптозащиты (рассматриваются в лекции 17)
- сканеры безопасности
- системы обнаружения вторжений,
- средства обеспечения аудита.

Средства построения VPN были рассмотрены нами в лекции 12. Остальные упомянутые средства рассматриваются в настоящей и начале следующей лекций.

16.1. Межсетевые экраны

Назначение и принцип действия межсетевых экранов

Межсетевой экран - это программный или программно-аппаратный компонент компьютерной сети, осуществляющий контроль и фильтрацию (отбрасывание) проходящего через него сетевого трафика в соответствии с заданными правилами.

Исходным англоязычным термином, применяемых повсеместно для именования *межсетевых экранов* (МЭ) является термин *firewall*, означающим вопреки предположениям не слишком глубоких знатоков английского языка совсем не “огненную стену”, а *противопожарную перегородку* между трюмами корабля, препятствующую распространению возможного пожара из охваченного им отсека трюма, в отсеки, огороженные от него противопожарными перегородками. Отметим, что в русскоязычной литературе МЭ иногда называют брандмауэром. Этот термин получен транслитерацией немецкого названия противопожарной перегородки, поэтому он ничуть не лучше (а хуже) транслитерации исходного термина - файрвол, являющейся более понятной подавляющему большинству сетевых специалистов и наиболее часто используемой в соответствующей литературе.

Метафора противопожарной перегородки положена в основу структурной организации компьютерных сетей, противостоящим недопустимым вторжениям извне. Компьютерные сети “отгораживаются” одна от другой такими противопожарными

перегородками - межсетевыми экранами, которые должны препятствовать недопустимым вторжениям в сеть, которые, однако, в отличие от пожара могут быть инициированы потенциально опасной активностью, исходящей как изнутри сети (например, как результат внедрения троянской программы на одном из компьютеров сети), так, в первую очередь, и из потенциально и реально агрессивного внешнего окружения, которым является интернет.

Таким образом, назначение МЭ состоит в ограничении состава нежелательных с точки зрения ИБ взаимодействий компьютеров из “закрытой” межсетевым экраном сети с компьютерами внешних сетей, находящихся за МЭ. При этом нежелательные взаимодействия характеризуются направлением пересылки пакетов (из защищённой внутренней сети во внешние или наоборот), а также адресной и, возможно, некоторой дополнительной информацией об источнике и получателе пакетов.

Естественным средством реализации рассмотренных ограничений является пакетный фильтр (точнее - пара фильтров), отбрасывающий (фильтрующий) нежелательные пакеты из потока пакетов, направляемых через МЭ из внутренней сети во внешние сети, и из потока пакетов, поступающих через МЭ из внешних сетей во внутреннюю сеть. В предыдущих лекциях мы рассматривали средство пакетной фильтрации, реализованное в маршрутизаторах. Напомним, что правила фильтрации пакетов, задаваемые при конфигурировании маршрутизаторов, задаются в виде списков контроля доступа ACL, вид которых рассмотрен в лекции о дополнительных функциях маршрутизаторов.

Средства пакетной фильтрации, реализованные в маршрутизаторах, фактически выполняют функции простейшего МЭ низшего уровня. В зависимости от того, какая информация протоколов каких уровней об источнике и получателе пакетов может быть использована МЭ при фильтрации, и сам МЭ может быть классифицирован, как работающий на уровне наивысшего из упомянутых протоколов.

Тогда МЭ, построенные на базе реализованных в работающих на сетевом (3-м) уровне маршрутизаторах средствах фильтрации пакетов, следует отнести к МЭ 4-го (транспортного) уровня, поскольку фильтрации выполняется на базе информации сетевого (IP-адреса источника и получателя пакетов) и транспортного (номера портов источника и получателя пакетов). На этом же уровне может дополнительно анализироваться и тип транспортного протокола.

Применение МЭ более высоких уровней позволяет выполнять при фильтрации более изощрённые проверки. Так МЭ 5-го (сеансового) уровня, называемые также *шлюзами сеансового уровня*, исключает прямое взаимодействие внешних компьютеров с компьютерами, расположенными в локальной сети, выступая в качестве посредника (*proxy*), который для каждого из входящих пакетов и проверяет их допустимость на текущей фазе соединения. При этом ни один сетевой пакет не пропускается через сеансовый шлюз (отфильтровывается), если он не является одним из пакетов установления соединения и не принадлежит ни одному из ранее установленных соединений.

МЭ прикладного (7-го), уровня могут анализировать информацию, передаваемую на уровне сетевых приложений, используя специальные приложения-посредники (*application proxy*), каждое из которых обслуживает свой прикладной протокол. С использованием соответствующих посредников, эти МЭ могут, например, отфильтровывать пакеты с командой PUT протокола FTP (предоставляя тем самым доступ к FTP-серверу только на чтение информации), выбрасывать пакеты почтового протокола SMTP с исполнимыми вложениями (которые потенциально могут быть заражены вирусами), выполнять аутентификацию пользователей, а также проверять, что SSL-сертификаты подписаны определённым удостоверяющим центром.

МЭ прикладного уровня разработаны для многих протоколов, включая, в частности, HTTP, FTP, почтовые (SMTP, POP, IMAP), TELNET и некоторые другие. Однако такая детальность проверок достигается ценою повышения вычислительной сложности выполняемой пакетной фильтрации, что влечёт либо потребность в реализации МЭ на высокопроизводительной вычислительной базе, либо невозможность использования таких МЭ для фильтрации трафика приложений, работающих в реальном времени.

Кроме того, чтобы МЭ прикладного уровня мог фильтровать трафик определённого сетевого протокола в него должна быть исходно включена или добавлена соответствующая программа прикладного proxy требуемого приложения. Дополнительно отметим, что в настоящее время попытка фильтрации данных протокола HTTP почти наверняка обречена на провал ввиду практически повсеместного применения криптографического протокола HTTPS вместо протокола HTTP: выполнять требуемую для фильтрации дешифрацию HTTPS трафика МЭ никоим образом не сможет, поскольку ему недоступны применявшиеся при шифровании различных потоков такого трафика закрытые ключи.

Вопросы расположения межсетевых экранов в структуре корпоративной сети

Межсетевой экран корпоративной сети обычно устанавливается в точке соединения этой сети с внешними сетями, обеспечивая требуемую конфигурацию изоляции между внутренними компьютерами и/или подсетями корпоративной сети и компьютерами и/или подсетями, находящимися извне этой сети. При этом, если межсетевой экран и пограничный маршрутизатор реализованы на различных сетевых устройствах необходимо решить вопрос о выборе варианта их взаимного расположения, каждый из которых имеет не рассматриваемые здесь достоинства и недостатки.

Однако, возможны и более сложные конфигурации. В частности, в составе корпоративной сети могут находиться серверы, доступ к которым извне принципиально не следует ограничивать. К таким серверам относятся, например: сервер официального сайта организации (являющийся “лицом” этой организации в интернете), FTP-сервер (содержащий открытые для всего сетевого сообщества документы и файлы с иными информационными и программными ресурсами) и некоторое другие виды открытых всему сетевому сообществу серверов. Очевидно, что “прокачивать” через фильтры МЭ трафик между такими серверами и внешним окружением сети совершенно нецелесообразно, поскольку такая фильтрация лишь создаёт дополнительную бесполезную вычислительную нагрузку на МЭ, вынуждая выполнять все проверки, определяемые в ACL МЭ, для заведомо допустимых пакетов трафика между такими серверами и внешними сетями.

Избавиться от таких ненужных проверок можно путём разнесения функций МЭ и пограничного маршрутизатора между различными физическими устройствами, расположения МЭ “за пограничным маршрутизатором” (по направления входа извне в корпоративную сеть), а также размещения упомянутых выше общедоступных серверов в специальной зоне корпоративной сети, находящейся между её МЭ и пограничным маршрутизатором и называемой демилитаризованной зоной.

Отметим также, что внутренние подсети корпоративной сети могут защищаться друг от друга не только путём оформления их в виде внутренних VPN, но и путём установки МЭ на входе из корпоративной сети в её внутренние подсети. И, наконец, МЭ может входить (и обычно входит) в состав операционных систем любых компьютеров сети. Таким

образом, защита компьютерной сети с помощью совокупности МЭ может иметь эшелонированный характер, как это показано на Рис. 16.1 (с учётом наличия демилитаризованной зоны).

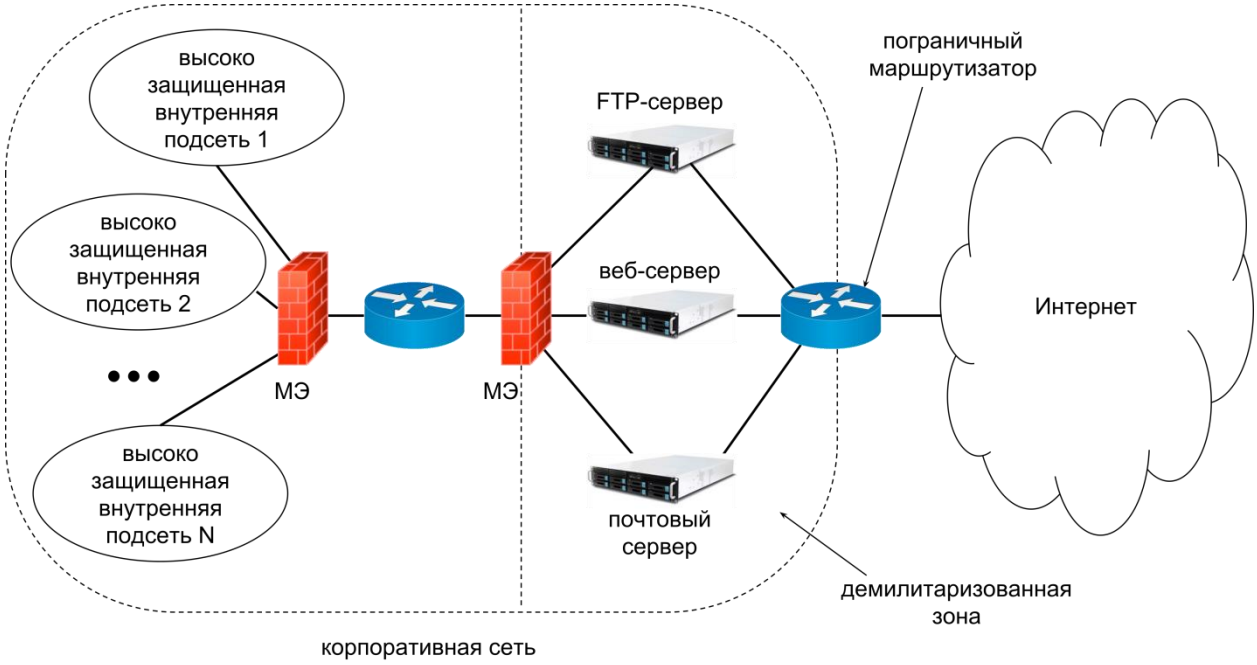

Рис. 16.1. Пример структурной организации системы защиты на базе множества МЭ

Технические способы реализации межсетевых экранов

Межсетевые экраны могут быть реализованы программно в составе ОС любых компьютеров сети, ОС маршрутизаторов, а также в виде ПО специальных серверов, основной функцией которых является межсетевое экранирование. Кроме того, МЭ могут быть реализованы в виде аппаратно-программных продуктов, иногда называемых чёрными ящиками МЭ (Black Box FireWall). Вкратце охарактеризуем МЭ, реализованные каждым из этих способов.

МЭ персональных компьютеров

Межсетевые экраны персональных компьютеров (ПК) или ПК-серверов (серверов, созданных на базе ПК) “охраняют” от нежелательного взаимодействия с окружающей сетевой средой сетевые приложения тех ПК, на которых они установлены. Такие МЭ подразделяются на встроенные в ОС и устанавливаемые дополнительно. Встроенные в ОС (например, в MS Windows) МЭ (файрволы) обычно обеспечивают минимум избирательности при ограничении допустимых взаимодействий работающих на них

приложений с внешней средой. Это позволяет полностью блокировать либо доступ к приложениям через определённые порты, либо всё взаимодействие определённых приложений со средой (одно приложение может взаимодействовать с сетевой средой через несколько портов. Даже возможности блокировать доступ не для всего сетевого окружения, а для определённой совокупности входящих в него индивидуальных компьютеров и подсетей такие МЭ обычно не предоставляют.

В силу указанных обстоятельств зачастую возникает потребность в установке на ПК дополнительного МЭ, обеспечивающего лучшую избирательность ограничений доступа, в том числе предоставляющего возможности фильтрации пакетов на прикладном уровне. Это предотвращает, например, проникновение в компьютер через сеть вирусов, троянов и пр. (в случае, если они “закачиваются” в компьютер при помощи не криптографических протоколов). Существует достаточно большое множество свободно распространяемых МЭ, предназначенных для установки на компьютерах, работающих под управлением MS Windows и обладающих достаточно развитыми средствами фильтрации пакетов на прикладном уровне. Одним из лучших (если не просто лучшим), МЭ рассматриваемого класса является **Comodo Firewall** – эффективный бесплатный МЭ, обеспечивающий очень высокий уровень защиты от сетевых угроз, достигаемый путём блокировки проникновения в компьютер вредоносного ПО и даже блокировки сетевых атак (методами, рассматриваемыми ниже) Comodo сканирует все процессы ОС и сетевые соединения при работе компьютера в сети и “посещения” пользователем этого компьютера потенциально опасных сайтов.

При обнаружении подозрительных операций, которые происходят при такой работе, Comodo в случае критически опасных действий блокирует их, информируя пользователя об обнаруженных угрозах и выполненных блокировках. И, наконец, Comodo имеет простой и интуитивно понятный интерфейс, благодаря чему он может быть рекомендован для применения даже неопытными пользователями, никогда ранее не работавшими с такими программами.

МЭ, создаваемые на базе маршрутизаторов

Простейшие МЭ, которые могут быть созданы на базе любых маршрутизаторов - это рассмотренные нами пакетные фильтры, анализирующие заголовки пакетов на 3-м и 4-м уровне сетевых протоколов (IP-адреса и номера портов прикладных процессов) и выполняющие фильтрацию пакетов на основе критериев, заданных списками доступа ACL.

Однако на базе маршрутизаторов некоторых производителей можно создать и МЭ более высоких уровней. В частности более чем широко известной компанией Cisco Systems разработано, как минимум, 2 таких решения: Cisco ASA (Adaptive Security Appliance - адаптивное устройство обеспечения безопасности) и Cisco IOS firewall. Второе решение является более продвинутым, но первое сертифицировано по линии ФСТЭК, поэтому перечислим основные возможности именно первого из упомянутых решений. К этим возможностям относятся, в частности: межсетевое экранирование с учётом состояния соединений и глубокий анализ протоколов прикладного уровня. Кроме того, Cisco ASA обеспечивает некоторые функции VPN-шлюза, поддерживая IPsec VPN соединения и SSL VPN соединения, обычно используемые при подключении к сети через веб-интерфейс. И, наконец, устройства Cisco ASA, как и все маршрутизаторы, обеспечивают маршрутизацию IP-пакетов, но применяя для управления маршрутизацией лишь протоколы внутренней маршрутизации RIP, EIGRP и OSPF, а также обеспечивают поддержку NAT.

Поэтому, хотя устройства Cisco ASA позиционируются производителем как аппаратные МЭ, они могут выполнять также функции VPN-шлюзов и функции обычных маршрутизаторов. Однако отсутствие в этих устройствах поддержки протокола управления внешней маршрутизацией BGP не позволяет использовать их в качестве пограничных маршрутизаторов AS, допуская при этом его установку на границе между защищённой частью AS и её демилитаризованной зоной.

Об автономных МЭ

Автономные МЭ могут базироваться на реальных серверах, на виртуальных серверах различных платформ виртуализации и на специальных устройствах, обычно называемых аппаратными МЭ или “черными ящиками” МЭ. Важной возможной особенностью аппаратных МЭ (не обязательно присущей всем аппаратным МЭ) является такая их реализация, при которой доступ “внутрь” таких МЭ (например, для контроля их работы или для изменения настроек выполняется лишь при помощи специальных средств, взаимодействующих с МЭ лишь по протоколам канального уровня либо надстроенных непосредственно над этим уровнем (а не над уровнем IP) протоколов. В этом случае МЭ не имеет собственных IP-адресов и прозрачно пропускает через себя весь разрешенный правилами фильтрации трафик. Такой МЭ не может быть подвергнут никаким атакам извне сети, поскольку все они могут быть реализованы только на уровнях начиная с IP и выше. И, будучи установленным на входе сети непосредственно перед пограничным

маршрутизатором этой сети, такой МЭ способен обеспечить экранирование такого пограничного маршрутизатора.

Отметим, что многие реализации МЭ обеспечивают также выполнение функций систем обнаружения вторжений (рассматриваются далее) и туннелирования (с криптозащитой) передаваемого трафика (то есть - функции VPN-шлюза). Для сетей, требующих обеспечения высокого уровня ИБ, МЭ должны быть сертифицированы по линии ФСТЭК, а, при предоставлении функции туннелирования трафика, и по линии ФСБ.

Средства, частично реализующие функциональность межсетевых экранов

Некоторые из ранее рассмотренных нами средств могут быть использованы для частичного выполнения функций МЭ. К ним относятся средства трансляции сетевых адресов NAT, кэширующие proxy-серверы (см. обсуждение вопроса “Кэширование”) и программы wrapper’ы (, обсуждение вопроса “Компрометация службы DNS”). Вкратце рассмотрим, какую функциональность МЭ обеспечивает каждое из указанных средств.

Средства трансляции сетевых адресов NAT

Транслятор сетевых адресов NAT, сконфигурированный на внешнем порте пограничного маршрутизатора некоторой сети, полностью скрывает её адресное пространство от всех внешних сетей и делает невозможным даже простое сканирование портов, скрытой за NAT’ом сети. Доступ извне к какому-то из компьютеров этой сети (кроме серверов, трансляция адресов которых выполняется через Dynamic NAT, и которые можно отнести к своеобразной демилитаризованной зоне) возможен только для тех серверов, к которым этот компьютер предварительно обратился с некоторым запросом.

А состав этих серверов может быть при необходимости ограничен в индивидуальном порядке локальными МЭ входящих в сеть компьютеров.

Эта схема может применяться в не требующих высокого уровня ИБ небольших сетях с пограничными маршрутизаторами, недостаточно мощными для выполнения вычислительно ёмкой фильтрации трафика по большим спискам контроля доступа.

Кэширующие proxy-серверы

Основным назначение таких серверов является ускорение доступа к часто используемым веб-страницам и снижение нагрузки на внешние каналы связи. Однако, если воспользоваться возможностью принудительного проксирования всего веб-трафика

некоторой сети (эта возможность реализуется путём перенаправления пограничным маршрутизатором сети всех пакетов с HTML-запросами, исходящих от внутренних компьютеров сети, на proxy-сервер этой сети), то все исходящие от компьютеров сети IP-пакеты с HTML-запросами будут содержать в качестве IP-адреса отправителя IP-адрес proxy-сервера, а все входящие в сеть IP-пакеты с HTML-страницами ответов на запросы будут содержать этот же адрес в качестве адреса получателя. И в случае простого прослушивания этих пакетов взломщику нельзя будет узнать IP-адреса других компьютеров сети. А в случае перехвата соединений с целью возможной атаки, атаковать получится только proxy-сервер, который, конечно же, должен быть серьёзно защищён. Учёт такого поведения может помочь сократить размер списков контроля доступа пограничного маршрутизатора и тем самым снизить вычислительную нагрузку на этот маршрутизатор.

Программы wrapper'ы (обёртки)

Как упоминалось ранее программы этого класса пропускают через себя все реализуемые посредством сетевых протоколов интерфейсы некоторых "оборачиваемых" wrapper'ом программ: от программ конкретных приложений до всей совокупности программ прикладного уровня. При этом каждый из пакетов, проходящий через такой, пропускаемый через wrapper интерфейс, может быть подвергнут дополнительной обработке wrapper'ом, например аутентификации отправителя пакета или фильтрации пакета по определённым правилам. Программы-wrapper'ы могут быть специализированными, предназначенными для совместной работы с некоторым конкретным приложением, или универсальными, "оборачивающими" всю совокупность сетевых прикладных программ.

Наиболее известным универсальным wrapper'ом является TCP Wrapper, доступный в среде большинства ОС семейства UNIX и ОС маршрутизаторов различных производителей. Наряду с некоторыми другими функциями TCP Wrapper обеспечивает возможность фильтрации IP-трафика на основе правил, заданных в виде традиционно используемых в пакетных фильтрах маршрутизаторов списков контроля доступа ACL, в которых вместо IP-адресов компьютеров могут указываться их доменные имена. Естественно, что на основе TCP Wrapper'а могут организовываться локальные МЭ UNIX-компьютеров и маршрутизаторов.

16.2. Сканеры безопасности

Сканер безопасности (security scanner) - это программные средства предназначенные для поиска возможных уязвимостей (брешей) в системе ИБ компьютеров сети. По месту их установки и зоне проверки (один компьютер или несколько (например, все) компьютеры некоторой сети) сканеры безопасности (далее - СБ) могут быть локальными (host based) или сетевыми (network based).

Локальный СБ может быть установлен на любом из компьютеров сети и анализировать наличие уязвимостей только в программном обеспечении того компьютера, на котором он установлен. Сетевой (network based) СБ устанавливается на одном из компьютеров и обычно может быть применён для анализа уязвимостей любого компьютера подсети, управляемой проводящим проверку сетевым администратором.

Что касается методов работы локальных и сетевых СБ, то локальные СБ могут проводить все проверки, выполняемые сетевыми СБ реализуя некоторые из них более эффективно, а также выполнять ряд дополнительных проверок, возможных благодаря наличию прямого доступа к памяти и файловой системе проверяемого компьютера и возможности выполнения анализа ИБ на уровне ОС проверяемого компьютера.

Методы работы сканеров безопасности

Рассмотрим общую организацию проведения анализа ИБ и методы такого анализа для сетевых СБ, делая специальные оговорки и дополнения, касающиеся локальных СБ. Общий сценарий работы практически любого СБ включает следующие основные этапы, часть из которых не выполняется некоторыми СБ:

1) Сбор информации о компьютерах проверяемой сети (или о локальном компьютере).
2) Обнаружение потенциальных уязвимостей.
3) Подтверждение действительного наличия обнаруженных уязвимостей.
4) Генерация отчетов об обнаруженных уязвимостях
5) Возможное автоматическое устранение обнаруженных уязвимостей.

Вкратце рассмотрим методы выполнения каждого из этих этапов

Сбор информации о компьютерах проверяемой сети

Для выполнения сбора такой информации проверяется наличие подключённых компьютеров по каждому из IP-адресов анализируемой сети и выполняется сбор

информации о запущенных на обнаруженных компьютерах сетевых демонах и их версиях. Для сбора этой информации выполняется сканирование портов проверяемого компьютера (попытка установления соединений с демонами всех портов) и анализ заголовочных пакетов, полученных от этих демонов. Большинство демонов передают инициатору соединения такие заголовки, содержащие, в частности, информацию о типе реализации и версии демона (для некоторых прикладных протоколов существует более одной реализации). Отметим, что демоны некоторых прикладных протоколов, например, HTTP, не направляют инициатору соединения заголовочных пакетов. Но, в случае HTTP, требуемая заголовочная информация может быть получена по запросу HEAD.

Для локальных СБ с установленной системой анализа защищенности на уровне операционной системы (например - System Scanner) требуемая информация может быть получена средствами этой системы.

Обнаружение потенциальных уязвимостей

На основе полученной на предыдущем этапе информации о работающих на различных компьютерах сканируемой сети сетевых демонах и версий этих демонов с использованием специальной БД сканера безопасности определяется, существуют ли известные уязвимости для каждого из упомянутых демонах и, если существуют, то какие именно. Для каждой из таких уязвимостей в БД СБ содержится информация о степени опасности уязвимости, о том, как воспользоваться уязвимостью для проведения атаки, возможная информация о способах устранения уязвимости и некоторая другая информация используемая на следующих этапах работы СБ.

По степени опасности уязвимостей они классифицируются на несколько уровней, различных для разных СБ. Так в СБ Internet Scanner все уязвимости в делятся на три степени опасности: высокая (high), средняя (medium) и низкая (low).

Отметим, что информация об обнаруженной по сведениям из БД уязвимости конкретной версии демона для конкретных версий демонов может быть недостоверной, особенно для свободно распространяемых в исходном коде демонов, поскольку администратор системы мог устранить уязвимость в коде программы, но не поменять её номер в заголовочной информации демона. Кроме того, отсутствие любой из уязвимостей в некоторой версии демона совсем не гарантирует её отсутствие в последующих версиях демона. И, если БД СБ ещё не обновлена и не содержит информацию о новой версии какого-либо демона, судить об отсутствии уязвимостей для этой версии демона на основе

их отсутствия в предыдущей версии можно лишь с определённой вероятностью, которая может быть определена методами математической статистики на основе информации о наличии различных уязвимостей во всех предыдущих версиях этого демона.

Отметим также, что локальные сканеры безопасности могут выполнять поиск уязвимостей в произвольных программах того компьютера, на котором они установлены, методами, подобными методам поиска вирусов, то есть путём поиска в кодах исполнимых всех программ фрагментов кода, сопоставимых с шаблонами уязвимого кода и называемого сигнатурами уязвимостей.

Подтверждение действительного наличия обнаруженных уязвимостей

Как отмечалось выше, информация о некоторых из уязвимостей, обнаруженных в результате анализа заголовков, полученных от различных демонов, может быть недостоверной, как в случае устранения уязвимости свободно распространяемого демона администратором системы. Кроме того, некоторые уязвимости, в том числе обнаруженные локальными СБ методами сигнатурного анализа кода программ, могут быть нейтрализованы, например действиями, выполняемыми МЭ соответствующих компьютеров. Поэтому необходимо каким-либо образом убедиться, что обнаруженные уязвимости действительно проявляются при работе соответствующих компьютеров в составе вычислительной сети.

И основным способом такой проверки является выполнение имитации атак (exploit check). При этом СБ направляет проверяемому компьютеру описанную в БД СБ последовательность атакующих пакетов для имитации атаки, соответствующей проверяемой уязвимости. И, если проверяемая уязвимость действительно имеет место, имитируемая атака должна “сработать”, вызывая соответствующий разрушительный эффект (см. ниже о мерах предосторожности). Отметим, что локальный СБ также может выполнять такую же имитацию произвольных сетевых атак, указывая в качестве IP-адреса получателя атакующих пакетов адрес обратной связи (loopback) 127.0.0.1.

Генерация отчетов об обнаруженных уязвимостях

На основе информации, собранной на предыдущих этапах, СБ генерирует отчёты с описаниями, обнаруженных уязвимостей. В некоторых СБ (например, в Internet Scanner)

создаются несколько отчётов, предназначенных различным категориям пользователей, начиная от сетевых администраторов и заканчивая руководителями ИТ-подразделений организации и даже её высшему руководству. При этом, если сетевых администраторов в первую очередь интересуют технические детали, то для руководителей разного уровня необходимо представить *наглядные* интегрированные отчёты, оформленные с применением графиков и диаграмм и с минимумом подробностей. Важной особенностью отчётов, генерируемых некоторыми СБ, является наличие в них рекомендаций по устранению обнаруженных проблем. По этой части безусловным лидером является СБ Internet Scanner, который для каждой уязвимости содержит пошаговые инструкции по устранению уязвимостей, специфичных для различных операционных систем.

Возможное автоматическое устранение обнаруженных уязвимостей

Этот этап очень редко реализуется в сетевых СБ, но часто применяется в системных сканерах, таких, например, как System Scanner. Указанная возможность может реализовываться различными способами. В частности, в System Scanner создается специальный сценарий “ремонта” уязвимости (fix script), который может быть применён системным администратором для устранения уязвимости. Отметим, что в системе System Scanner одновременно с созданием сценария fix script, создается и “обратный ему” сценарий, производящий “откат” от “исправленного” состояния содержавшей уязвимости системы в исходное состояние. Такая возможность может быть востребована в случае, если после устранения уязвимости нормальное функционирование компьютера оказалось нарушено. В других известных нам системах такого типа возможности автоматического "отката" к “доремонтному состоянию” не предусмотрено, но при необходимости такой “откат” может быть выполнен за счёт применения результатов полного резервного копирования состояния всей системы ПО компьютера непосредственно перед выполнением имитации атаки.

О существующих реализациях сканеров безопасности

В настоящее время доступен весьма широкий ряд различных СБ, как доступных лишь коммерчески, так и свободно распространяемых. Некоторые из них, такие как неоднократно упоминавшийся выше Internet Scanner фирмы ISS является универсальным в том смысле, что он может обнаруживать уязвимости в сетевом ПО различных ОС (включая различные ОС семейства UNIX, семейства MS Windows и др.), обеспечивающих

взаимодействие работающих под их управлением компьютеров в сетях, основанных на применении сетевых протоколов семейства TCP/IP.

Другим универсальным коммерчески доступным СБ, признанным одним из лучших СБ, является Nessus.

Наряду с широким кругом универсальных СБ, существует достаточно широкий ряд СБ “заточенных” на определённые ОС или классы ОС. Узнать их более или менее полный перечень можно при помощи поисковых запросов к интернет примерно следующего вида: “сканер безопасности для MS Windows”.

Ещё одна разновидность СБ ориентирована на обнаружение уязвимостей в веб-приложениях, как наиболее широко распространённых в настоящее время сетевых приложений. В качестве яркого представителя этой разновидности можно привести сканер поиска веб-уязвимостей METASCAN, сам оформленный как веб-приложение и являющийся доступным через любой современный браузер инструмент по поиску различных уязвимостей сайтов (например атаки из списка Owasp top 10, XSS-атаки, SQl-инъекции и другие), включая проверку уязвимостей CMS и плагинов. При этом METASCAN позволяет проводить поиск уязвимостей сайтов, созданных в среде самых разных ОС.

И в завершение нашего более чем беглого обзора сканеров безопасности упомянём универсальный (обнаруживающий уязвимости для широкого круга сетевых ОС) СБ XSpider, сертифицированный ФСТЭК. Этот сканер имеет как свободно распространяемые, так и коммерческие версии отечественных производителей, стоимость которых может варьироваться в достаточно широких пределах. По утверждениям некоторых интернет-источников он является лучшим СБ, широко используемым в России за последние 15 лет. Другие источники утверждают, что XSpider - это “единственный в мире сканер, уже сегодня определяющий более трети уязвимостей, которые принесет завтрашний день”. Но безотносительно к двум последним дифирамбам отметим, что, во-первых, благодаря наличию бесплатных и относительно недорогих (примерно от 9000 руб.) реализаций этого сканера он может применяться малобюджетными организациями. А, во-вторых, наличие сертификата ФСТЭК для некоторых коммерческих реализаций XSpider’а позволяет использовать эти реализации для проверки уязвимостей сетей организаций, предъявляющих достаточно высокие требования к обеспечиваемому уровню ИБ.

О правилах применения сканеров безопасности

Главным правилом применения сканеров безопасности является правило недопустимости их применения к компьютерам, не входящим в зону ответственности системного администратора, выполняющего проверку сети. Несмотря на то, что большинство СБ могут выполняться в “облегчённом” режиме, предусматривающих выполнение лишь этапов 1,2 и 4 и не выполняющих этап 3, на котором проводится имитация сетевых атак, даже на этапе 1 выполняется такое действие как сканирование портов компьютеров проверяемой сети. А это действие классифицируется как возможное выполнение разведывательной стадии большого множества сетевых атак, то есть, как начальная стадия атаки. Поэтому применение СБ к компьютерам, не входящим в зону ответственности выполняющего это действие сетевого администратора попадает под действие ст. 273 УК РФ "Создание, *использование* и распространение вредоносных программ для ЭВМ”, в соответствии с которой обвиняемый может получить реальный срок лишения свободы.

Но и при применении СБ к компьютерам из своей зоны ответственности системный администратор должен предпринимать минимально необходимые меры предосторожности, позволяющие избежать неустранимых “разрушений” в ПО и БД проверяемых компьютеров и другими способами нарушить любой из аспектов ИБ систем, работающих с использованием проверяемых компьютеров.

Эти минимально необходимые меры включают следующие:

1) Применение СБ разрешено лишь в периоды времени, в которые допустимы временные нарушения работоспособности систем, работающих с использованием проверяемых компьютером. Иначе возможные временные отказы в обслуживании могут сказаться очень и очень отрицательно.
2) Перед применением СБ к какому либо из компьютеров в “боевом” режиме, предусматривающем выполнение этапа имитации атак, необходимо создать резервные копии всей системы ПО, включая ОС, все БД и всё прикладное и системное программное обеспечение. В случае, если хотя бы одна из имитируемых атак сможет оказать свой вредоносный разрушительный эффект, этот эффект может быть полностью нейтрализован путём загрузки созданной резервной копии системы.
3) При определении времени, требуемого на сканирование той или иной совокупности компьютеров необходимо суммировать время, необходимое для

создания резервных копий (поэтому эти копии нельзя делать в период активной работы системы, характеризующийся постоянным изменением БД и другой информации), время проведения сканирования и время, необходимое для восстановления систем всех подвергаемых проверке компьютеров из из резервных копий.

Отметим, что полное или частичное невыполнение этих правил может привести к временному, иногда весьма продолжительному выходу из строя атакованных при выполнении 3-го этапа сканирования их безопасности компьютеров и, как следствие, к длительному отказу в обслуживании для функций, выполняемых этими компьютерами. А это, в свою очередь, может привести к существенным финансовым и/или репутационным потерям организации, владеющей проверяемой СБ сетью. А если такие события произошли, то проводившему проверку ИБ сетевому администратору может быть предъявлено обвинение в нарушении ст. 274 УК РФ "Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети", при доказательстве которого в суде сетевому администратору может быть назначено уголовное наказание.

Отметим также, что в дополнение к рассмотренным правилам существует ряд других, менее важных правил, рассмотрение которых выходит за рамки настоящего учебного курса.

16.3. Системы обнаружения вторжений IDS

Система обнаружения вторжений (Intrusion Detection System - *IDS*) - это система, обнаруживающая факты проведения разнообразных сетевых атак, от сравнительно “безобидного” простого сканирования портов входящих в сеть компьютеров до предельно широкого множества атак, производящих существенный вредоносный эффект. В отличие от МЭ IDS не фильтрует пакеты, поступающие из потенциально опасных (но не обязательно опасных) источников, а анализирует последовательность поступающих пакетов, выявляя в них подпоследовательности пакетов, реализующих атакующие воздействия, и, зачастую (т.е. во многих IDS), при достижении атакующим воздействием некоторой “критической черты” автоматически блокирующей опасное атакующее сетевое соединение. IDS, обеспечивающие последнюю возможность фактически являются системами *обнаружения* и *предотвращения* вторжений. Однако, мы не будем выделять их в отдельный класс и всегда будем называть их “IDS”.

Основные классы IDS по месту их расположения в защищаемой сети

По по месту их расположения в защищаемой сети IDS можно выделить несколько основных их “базовых” классов. При этом конкретная IDS может реализовывать возможности нескольких базовых классов. Вкратце рассмотрим эти классы.

- ***Сетевые*** (network based) IDS обнаруживают атаки, анализируя последовательности пакетов, передаваемые через некоторый канал сети. При этом обычно анализ выполняется для копии сетевого трафика реально используемого в сети канала, получаемой, например, путем зеркалирования этого сетевого трафика на порте коммутатора или маршрутизатора. Выполнение анализа на копии трафика необходимо из соображений недопустимости внесения каких-либо задержек в поток реального трафика сети: выполнение параллельного с передачей реального трафика анализа копии этого трафика (возможно, на другом компьютере) никак не сказывается на скорости прохождения реального трафика. Отметим, что сетевая IDS может быть установлена и на единичный компьютер с целью обнаружения атакующих последовательностей пакетов, поступающих с сетевого интерфейса этого компьютера. При этом, как правило, поток пакетов, направляемый для IDS, является копией потока пакетов, поступающих с сетевого интерфейса, получаемой с использованием виртуальных сетевых драйверов TUN/TAP, доступных в большинстве сетевых ОС.
- ***Локальные*** (host based) IDS или ***локальные агенты*** сетевых IDS расположены в составе ОС компьютеров и обнаруживают факты атакующих вторжений, осуществляя анализ выполняемых в ОС системных вызовов, изменений системных журналов ОС и приложений, выполняемых модификаций файлов определённых типов (исполнимых файлов, файлов паролей, системных баз данных и др), изменений состояния изменения состояния различных внутренних структур данных ОС и пр.
- **Основанные на коммуникационном протоколе** (protocol based) IDS, являются локальными системами или агентами сетевых IDS. Они анализируют трафик, передаваемый (в 2-х направлениях) между локальным компьютером и компьютерами, взаимодействующими с локальным, по определённым высокоуровневым коммуникационным протоколам. Так для веб-сервера подобная IDS обычно анализирует трафик, передаваемый по протоколам HTTP и HTTPS. При этом для контроля использовании трафика, передаваемого по HTTPS, IDS должна получать копию этого трафика с такого внутреннего интерфейса веб-сервера, через

который содержимое HTTPS пакетов передаётся ещё до их шифрования перед отправкой в сеть.

- ***Основанные на прикладном протоколе*** (application protocol based) IDS, являются локальными системами или агентами сетевых IDS. Они анализируют потоки данных, передаваемых с использованием протоколов прикладного уровня, специфичных для определенных приложений. Например, на веб-сервере с SQL базой данных такие IDS могу анализировать содержимое SQL команд, передаваемых на сервер.

Легко заметить, что 3 последних класса IDS (кроме сетевых), фактически могут использоваться либо как локальные IDS, либо как агенты других, очевидно, сетевых IDS, которые могут включать агентов IDS 3-х последних классов в качестве своих подсистем.

Методы обнаружения атак, применяемые в сетевых IDS

Рассмотрение совокупности методов, применяемых в различных разновидностях локальных IDS, требующее достаточно детальных знаний по методам реализации операционных систем, веб-серверов и пр., выходит за рамки настоящего учебника, в котором мы ограничимся лишь рассмотрением методов работы сетевых IDS.

При грубой классификации методов, применяемых в IDS для обнаружения атак, их разбивают на сигнатурные (signature based); основанные на анализе аномалий (anomaly-based) и называемые также поведенческими (behavioral based); основанные на правилах (rule based) и некоторые другие. Конкретные IDS могут поддерживать одновременное (параллельное во времени) применение нескольких методов.

Но во всех практически применяемых IDS основной “рабочей лошадкой” являются сигнатурные методы, которые мы и рассмотрим значительно подробнее, чем остальные.

Сигнатурные методы обнаружения сетевых атак

Сигнатурные методы обнаружения сетевых атак основаны на поиске в последовательности входящих пакетов совокупности фрагментов этой последовательности, сопоставимой с некоторым шаблоном “атакующей” последовательности пакетов. Каждый из таких шаблонов называется сигнатурой атаки. И здесь имеет место полная аналогия с сигнатурными методами обнаружения вирусов, в которых фрагменты анализируемого на наличие в коде анализируемой программы

фрагментов, сопоставимых с сигнатурами вирусов. И в том и в другом случае сигнатура (атаки или вируса) описывается некоторым регулярным выражением, сформулированным на некотором языке регулярных выражений. Но эти языки отличаются так, чтобы язык описания сигнатур вирусов позволял адекватно описывать шаблоны последовательностей бит кода программы, а язык описания сигнатур атак - шаблоны атакующих последовательностей пакетов. В качестве известного если не всем, то подавляющему большинству читателей настоящей книги языка регулярных выражений, можно указать язык описания шаблонов имён файлов, используемых в утилитах массовых операций над файлами в командной строке ОС семейства UNIX и MS Windows.

Причина, по которым сигнатуры описываются именно регулярными выражениями, состоит в том, что при их использовании для описания шаблонов, процесс сопоставления анализируемого фрагмента с сигнатурой (шаблоном фрагмента) может быть выполнен за один проход по тексту шаблона и сопоставляемого фрагмента. А это значит, что алгоритм сопоставления имеет линейную сложность: время нахождения сопоставимого с сигнатурой фрагмента пропорционально длине этого фрагмента. А это, в свою очередь, означает, что при согласованной скорости работы процессора компьютера, на котором работает IDS, со скоростью работы канала, трафик которого анализируется на наличие атакующих последовательностей пакетов, поиск атакующих последовательностей можно вести в реальном времени, в темпе поступления анализируемых пакетов, что чрезвычайно важно, например, для обеспечения возможности оперативной блокировки источника атаки.

Перейдём к более детальному рассмотрению организации процесса сопоставление потока входящих пакетов с сигнатурами атак. При этом в первую очередь отметим, что последовательности пакетов, реализующих атаки разных типов, могут быть подпоследовательностями либо всего интегрированного потока пакетов (как, например, для сканирования портов), либо любого из индивидуальных соединений пары взаимодействующих через сеть процессов, идентифицируемых IP-адресами и номерами портов двух “конечных точек” индивидуального соединения. К атакам первого типа относятся, как правило, более или менее универсальные разведывательные действия типа сканирования портов, проводимые для сбора информации, необходимой для определения применимости конкретных атак для выбранного объекта атаки. К атакам второго типа относятся атаки, направленные на сетевые демоны того или иного типа. При этом, поскольку и сканирование портов различных подсетей защищаемой посредством

IDS сети и атаки на различные сетевые демоны должны обнаруживаться параллельно, то выполнение такой параллельной (точнее - асинхронной) обработки должно осуществляться совокупностью асинхронных процессов (классических полновесных процессов или легковесных процессов-нитей - threads), далее называемых просто “процессами”. Поэтому интегральный процесс поиска во входном потоке атакующих последовательностей пакетов организован как совокупность процесса анализа интегрированного входного потока и процессов анализа индивидуальных потоков для всех активных сетевых соединений, проходящих через анализируемый канал.

Если при этом протокол какого-либо из соединений подвержен нескольким атакам (имеющим разные сигнатуры), то для выявления в этом соединении фрагментов, сопоставимых с каждой из сигнатур в реализации IDS, должно быть предусмотрено порождение отдельного процесса (скорее всего - процесса-нити).

Описание сигнатур всех атак, поиск сопоставлений которых выполняется совокупностью рассмотренных выше процессов содержатся в БД IDS, которая оперативно пополняется описанием новых сигнатур при выявлении новых типов атак и разработки сигнатур этих атак.

Описание каждой сигнатуры содержит определяющее соответствующий шаблон регулярное выражение, информацию о номере порта транспортного протокола, для которого возможна соответствующая атака (либо информация о применимости сигнатуры к интегрированному потоку), информацию о требуемом способе реакции на обнаружение сопоставимой с сигнатурой последовательностью пакетов, название атаки на языке, понятном людям и некоторая другая информация.

При рассмотренной организации сигнатурного обнаружения атак гарантируется:

- обнаружение любой из описанной в БД сигнатур атак в реальном масштабе времени
- обнаружение не только факта атаки, но и её типа и источника (определяется по сигнатуре), а также объекта применения атаки (по IP-адресу и номеру порта всех пакетов атакующей последовательности)

Обеспечение выполнения этих условий является большим достоинством сигнатурных методов обнаружения атак, не обеспечиваемым другими методами.

Но сигнатурный метод обладает и одним большим недостатком - он не способен обнаружить ни одну атаку, описания сигнатуры которой нет в БД IDS. Поэтому в состав многих IDS наряду с обязательными сигнатурными средствами обнаружения сетевых атак

входят как неотъемлемая часть и средства обнаружения атак одним или несколькими поведенческими методами, позволяющие в той или иной степени компенсировать указанный недостаток сигнатурных методов.

Краткие сведения о других методах обнаружения атак

Основанные на анализе аномалий (anomaly based) или поведенческие (behavioral based) методы обнаружения атак включают как подмножество методы, основанные на статистических аномалиях (statistical anomaly based). Такие методы базируются на использовании нормального профиля трафика (некоторого множества параметров этого трафика), создаваемого в специальном режиме достаточно продолжительного обучения (для получения нормальных статистических значений всех параметров профиля) на тщательно защищённой от всевозможных атак сети. Наличие атак на этапе обучения основанных на этом методе IDS приведёт к тому, что аномальный эффект этих атак будет учтён при вычислении нормального профиля, что в свою очередь приведёт к квалификации происходившей при обучении и повторно выполняемой при эксплуатации атакующей деятельности как абсолютно нормальной. После создания профиля, весь поступающий трафик сравниваются с этим профилем с использованием сложных статистических алгоритмов и присвоением рейтинга аномальности каждому пакету. Все существенные отклонения от профиля (достаточно высокий рейтинг аномальности одного или нескольких пакетов) считается атаками, при обнаружении которых предпринимаются соответствующие действия (см. ниже). Аналогичные методы могут применяться также в локальных IDS и локальных агентах сетевых IDS применительно к определению аномальных (атакующих) действий пользователей.

К числу других методов обнаружения атак относятся и так называемые методы, основанные на правилах (rule based), основанные на методах искусственного интеллекта, весьма далёких от направления нашего учебника и не упоминаемые нами далее.

Основным достоинством перечисленных методов по сравнению с сигнатурными является то, что этими методами в принципе можно обнаруживать факты проведения сетевых атак *ранее не известных типов*. При этом, однако, возможны ложные срабатывания системы, особенно в случае заниженного порога реагирования на величину рейтинга аномальности анализируемых пакетов. Благо, что при обнаружении атак этими методами IDS не располагает никакими возможностями автоматической блокировки атак. Ведь для атаки неизвестного типа эти методы не могут предоставить какую-то

информацию ни о том, каков механизм атаки (для сигнатурных атак каждой сигнатуре соответствует название атаки, данное для конкретного механизма её проведения), ни о том, какая последовательность пакетов осуществляет реализацию этой атаки, ни, наконец об источнике атакующей последовательности пакетов. Кроме того, эти методы могут иметь столь высокую вычислительную сложность, что анализ пакетов не может быть выполнен в реальном масштабе времени и фактически выполняется с некоторым опозданием. А пытаться заблокировать уже свершившуюся атаку просто невозможно.

О возможных способах реакции на обнаруженные атаки

При обнаружении атаки IDS может в качестве реакции на обнаруженную атаку выполнить одно или несколько следующих действий.

1) Журнализация сведений об обнаруженной атаке
2) Выдача в специальный файл структурированных сведений о состоянии ОС атакованного компьютера (локальными IDS или локальными агентами IDS)
3) Выдача в специальный файл дампа оперативной памяти атакованного компьютера (“посмертная выдача” полного состояния памяти атакованного компьютера при обнаружении факта проведения неизвестной атаки, выполненного соответствующими методами)
4) Отправка сообщения об обнаруженной атаке системным администраторам
   а) по электронной почте
   б) по SMS
5) Выполнение разрыва атакующего соединения с атакующим компьютером (автоматическое определение атакующего компьютера возможно только для сигнатурных IDS)
6) “Включение” блокировки источника атаки фильтрами МЭ (автоматическое определение источника атаки возможно только для сигнатурных IDS)
7) Немедленное уничтожение (команда или системный вызов kill) определённых процессов в ОС атакованного компьютера (возможно только при наличии на компьютере локальной IDS или локального агента IDS)
8) Перезагрузка ОС атакованного компьютера (возможно только при наличии на компьютере локальной IDS или локального агента IDS)

Перечисленные действия указаны в порядке возрастания опасности обнаруженной ситуации, в зависимости от степени этой опасности и методов, которыми она была обнаружена, выполняются одно или несколько из указанных действий.

В сигнатурных IDS информация о выполняемых при возникновении атаки действиях хранится в БД сигнатур атак в составе информации о каждой сигнатуре. При этом в состав выполняемых действий могут быть включены действия 1, 2 и 4-6 (действие 1 обычно выполняется и для всех других типов IDS.

Действия 2-3 обычно выполняются при обнаружении атаки неизвестного типа и предназначены для анализа возможной сути атаки и разработки её сигнатуры а также для определения источника атаки.

И, наконец, необходимость выполнения действий 7-8 обычно определяется локальными IDS или локальными агентами сетевых IDS. Эти действия (особенно 8-е) выполняются только в том случае, если продолжение работы определённых процессов атакованного компьютера или даже компьютера в целом грозит недопустимыми последствиями.

Краткие сведения о некоторых известных IDS

В первую очередь упомянем очень популярную свободно распространяемую сетевую IDS Snort. Не смотря на бесплатность этой системы, она хорошо поддерживается и её база данных постоянно и весьма оперативно пополняется описаниями сигнатур вновь обнаруженных атак. Если Вы являетесь сетевым администратором сети организации, политика ИБ которой не требует сертификации средств обеспечения ИБ, первой из рекомендуемых для Вас IDS является именно Snort.

Если политика ИБ Вашей организации предписывает Вам использовать лишь IDS, сертифицированные ФСТЭК, то, обратившись к поисковой системе, Вы получите довольно длинный перечень таких IDS. Мы упомянём лишь 2 таких IDS: Форпост и упоминавшийся при рассмотрении межсетевых экранов программно-аппаратный комплекс UserGate.

Система Форпост выявляет атаки на основе сбора и анализа информации о передаваемых пакетах данных на сетевом, транспортном и прикладном уровнях стека протоколов TCP/IP. Это позволяет, в частности, обеспечить возможность выявления атак, реализуемых нарушителем по криптозащищённым сетевым соединениям. IDS Форпост отслеживает и блокирует атаки в режиме реального времени с помощью эвристических правил и анализа сигнатур известных атак. В своей работе IDS Форпост применяет

специальные агентские программы, включающие так называемые сетевые и серверные датчики а также модули агенты. Совокупность этих средств обеспечивает защиту определённых сетевых сервисов и сетевых сегментов “охраняемых” этой IDS сетевой информационной системы. Используемыми в Форпост’е способами реакции на обнаруженные атаки являются: оповещение через консоль системы её администратора информацией об обнаруженной атаке; аварийное завершение сетевого соединения с источником атаки; блокирование возможности дальнейшего сетевого доступа к “охраняемой” сети со стороны источника атаки; запись информации об обнаруженной атаке в базу данных системы.

Выполняющая основную функцию МЭ система UserGate предоставляет также функции IDS. UserGate IDS отслеживает и блокирует атаки в режиме реального времени с помощью эвристических правил и анализа сигнатур известных атак. Администратор этой системы может создавать различные профили IDS (наборы сигнатур, релевантных для защиты определенных сервисов) и задавать правила IDS, определяющие действия для выбранного типа трафика (IP, ICMP, TCP, UDP), который в дальнейшем будет проверяться IDS в соответствии с назначенными профилями. Способами предотвращения развития обнаруженных атак являются блокирование доступа со стороны определённых подсетей, обрыв и принудительный разрыв атакующих соединений. Об указанных событиях и действиях оповещается администратор сети и заносится соответствующая информация в специальный системный журнал. При этом поскольку в состав UserGate входят одновременно и МЭ и IDS, требуемая для IDS блокировка доступа со стороны атакующих сетей легко и просто реализуется фильтрами МЭ, входящего в состав UserGate.

16.4. Краткая характеристика средств аудита ИБ

Общие сведения о рассматриваемых средствах аудита

Аудит информационной безопасности КСС - это процесс перманентной (никогда не завершающейся) оценки характеристик текущего состояния информационной безопасности КСС, выполняемый в соответствии с определёнными целями, критериями и, как правило, с применением соответствующих программных средств.

В рамках настоящего курса нас интересуют средства аудита, направленные на выявление не обнаруженных рассмотренными выше средствами других типов нарушений (обхода) системы безопасности КСС (например, деятельности “следов” и результатов злоумышленников при проведении ими внутренних атак), а также с целью расследования причин и поиска исполнителей реально состоявшихся атак на КСС, как сумевших “пробить” существующую систему защиты КСС, так и заблокированных этой системой защиты на некоторой стадии своего исполнения. При этом мы совершенно не затрагиваем далеко выходящих за пределы настоящего курса вопросов, связанных как с другими целями проведения аудита, так и с процедурой его проведения.

Средства поддержки аудита ИБ, применяемые в рассмотренных выше целях как правило основаны на ведении специальных *системных журналов* (называемых лог-файлами или просто логами – log), в которых фиксируется информация о действиях и/или событиях выполненных/произошедших в различных подсистемах КСС. И, совершенно естественно, что такие средства включают собственно средства журнализации, обеспечивающие оперативное занесение информации в упомянутые журналы, и средства избирательной выдачи интересующей аудитора информации из указанных журналов.

Отметим, что при такой организации средств аудита возможности расследования как следов и результатов деятельности внутренних взломщиков, так и причин “пробивания” внешними атаками системы ИБ КСС и поиска их исполнителей естественно ограничены тем, каков объем и качественный состав информации о функционировании КСС зафиксирован в совокупности системных журналов, применяемых этими средствами. А для хранения совокупности сколь либо широкого ряда объёмных системных журналов, содержащих достаточно детальные сведения о каждом из фиксируемых в этих журналах событий на протяжении достаточно большого времени (от нескольких дней до нескольких месяцев или лет) требуются большие объёмы внешней памяти. Кроме того, процесс занесения сведений в эти журналы в некоторой степени “тормозит” работу системы и влечёт повышение нагрузки и на процессоры “вовлечённых” в журнализацию

компьютеров и на их внешние запоминающие устройства. Процесс поиска в системных журналах интересующей информации с возрастанием количества и степени информативности записей системных журналов также связан с увеличением времени, необходимого для проведения такого поиска. Но, с другой стороны экономия на составе применяемых журналов недопустима для систем, требующих высокого уровня их ИБ. Поэтому вопрос определения того, журналы каких событий следует вести и насколько подробно фиксировать в этих журналах сведения о соответствующих событиях на самом деле является вопросом выбора компромисса между дополнительными затратами на требуемое для обеспечения возможности детального аудита оборудования и "разрешающей способностью" средств обнаружения всей информации, необходимой для возможности проведения *достаточно детального* расследования.

Естественно, что для КСС, предъявляющие сколь-либо высокие требования к уровню их ИБ, термин "возможность достаточно детального" расследования является слишком неопределённым. Поэтому уже на достаточно раннем этапе развития информационных систем, построенных с использованием компьютерных сетей (КСС, в терминологии нашей книги) появились документы, классифицирующие уровни возможной ИБ КСС и формулирующие минимальные требования к системам аудита, способных провести обнаружение и/или расследование инцидентов нарушения системы ИБ КСС, которые обязательно должны обнаруживаться и "нейтрализовываться" для обеспечения соответствующего уровня ИБ.

Первым из таких документов стала известная "Оранжевая книга" разработанная в 1970-х годах министерством обороны США и содержащая двухуровневую классификацию уровней ИБ и описание требований к возможностям соответствующих каждому уровню средств аудита. Верхние уровни этой двухуровневой классификации обозначаются несколькими 3-мя латинскими буквами (A, B и C), буквы, отстоящие дальше от начала алфавита соответствуют более требовательному к уровню необходимой ИБ классу систем. Подуровни этих уровней идентифицируются цифрами, так что б'ольшие цифры идентифицируют более высокие уровни ИБ. При этом уровень C2 был признан и признаётся до сих пор, уровнем ИБ, достаточным для банковских систем. В качестве кратко сформулированных требований к средствам аудита уровня C2 обычно приводят следующие: эти средства аудита должны предоставлять возможность выяснить для каждого поля базы данных системы с уровнем безопасности C2 когда (дата и время) и от имени какого пользователя было выполнено изменение этого поля, а также какое

значение это поле имело до этого изменения. При этом это требование должно пониматься в широком смысле и касаться *всех полей данных*, характеризующих состояние защищенной по уровню С2 системы, а не только полей данных, хранимых под управлением входящих в состав этой системы СУБД.

Перейдём к краткому (а порою и просто беглому) рассмотрению некоторых известных систем аудита. Отметим, что такие средства можно разбить на 3 существенно различных класса:

- Базовые средства аудита операционных систем
- Средства аудита различных приложений
- Комплексные системы аудита ИТ-инфраструктур (в нашей терминологии - КСС в целом)

Вкратце охарактеризуем некоторые системы каждого из упомянутых классов

Базовые средств аудита операционных систем

К средствам этого класса, отнесём средства аудита, поставляемые в составе дистрибутивов ОС. К их числу могут относиться как разрозненные средства ведения определённых системных журналов, так и пакеты с более или менее широкой функциональностью. В качестве примера таких средств, рассмотрим применяемый в составе некоторых разновидностей ОС UNIX отдельный системный журнал `/var/log/wtmp` (или `/var/log/utmp`), команду `lastcomm` относительно простого пакета аудита `acct` (или `pacct`), также применяемый в ряде разновидностей UNIX. И, поскольку более развитые средства аудита разных разновидностей UNIX достаточно сильно отличаются друг от друга - рассмотрим возможности средств аудита ОС Linux, реализованные в составе версии .

В ***системный журнал /var/log/wtmp*** в оперативном режиме заносится информация (в текстовом виде) о начале и завершении терминальных сеансов работы пользователей (в том числе - сеансов работы через удалённый терминал), включающая в частности, время события, логин-имя пользователя, идентификатор терминала и некоторую другую информацию. Доступ к содержимому этого файла может выполняться с использованием произвольных команд (утилит командной строки) системы UNIX. Средства более избирательного доступа могут быть получены путем применения широко используемого системными администраторами UNIX принципа “сделай сам”, поддерживаемого гибким командным языком UNIX (`sh` и его разновидностей) и простым, но мощным набором обычно применяемых в конвейерах команд-фильтров этого языка

(фильтрами в UNIX называются программы, выполняющие некоторое преобразование своего стандартного входного файла в стандартный выходной. В качестве примера эффективности возможного использования этих средств упомянём известную авторам разработку системы учёта времени доступа Dial-up пользователей (модемных пользователей) к сети (с различными “весами” для дневных и ночных часов работы), выполненную в середине 1990-х годов исключительно на базе упомянутых средств буквально за пару дней.

Выдача строк файла `/var/log/wtmpв` в свой стандартный выходной поток выполняется командой `last`

Отметим, что в журнале (и во всех других системных журналах) хранится информация лишь за последние сутки. Но обычно и для этого и для других системных журналов обеспечивается хранение в системе копий этого файла, созданных в каждых из N предыдущих суток и хранимых под именами `/var/log/wtmp1` – `/var/log/wtmpN` соответственно. Процесс переименования текущих копий этих файлов в файлы, соответствующие предыдущим суткам и называемый ротацией системных журналов обычно запускается соответствующим демоном (cron) раз в сутки в запланированной время (обычно ночное). Аналогичным образом обычно выполняется ротация и всех других системных журналов.

Команда `lastcomm` пакета `acct`, будучи вызвана без параметров, обеспечивает выдачу в свой стандартный выходной поток совокупность текстовых записей (строк), содержащих сведения о всех командах, выполненных пользователями системы. Каждая строка такого файла содержит имя выполненной команды (утилиты), логин-имя выполнившего её пользователя, информацию о терминале пользователя (возможно - терминале удалённого доступа), дате и времени выполнения команды. А пользуясь упомянутым принципом “сделай сам” системный администратор легко может написать, например, sh-скрипт, который по выходному потоку рассматриваемой команды подготовит весьма востребованный при аудите внутренних атак отчёт о том, какие пользователи и с каких терминалов выполняли команды из некоторого заданного параметром скрипта “джентльменского набора потенциально опасных команд” (включающего, как минимум, команду `su`, обычно применяемую для перехода в режим суперпользователя `root`), и ,при необходимости - в “неурочные» интервалы времени, заданные другим параметром скрипта.

Подсистема аудита системных событий ОС Linux. Рассматриваемые здесь средства были включены в ядро Linux начиная с версии 2.6. Подсистема аудита обеспечивает возможности отслеживания ***различных системных событий***, критичных с точки зрения ИБ. К ним относятся следующие (и некоторые дополнительные) типы событий:

- Выполнение операций чтения и записи в файлы, а также изменение прав доступа к файлам; таким образом журнализируются, например, и факты потенциально опасной установки бита SetUID для исполнимых файлов
- Установление и разрыв сетевых соединений
- Выполнение системных вызовов и сигналов ядру ОС
- Запуск и остановка приложений (любая команда-утилита также является приложением, так что эта возможность покрывает все возможности `lastcomm`)
- Попытки безуспешной авторизации в системе
- Изменение сетевых настроек
- изменение информации о пользователях и группах пользователей.

Поскольку все из указанных событий могут произойти лишь посредством использования для этого соответствующих системных вызовов (обращений к ядру ОС), то для отслеживания этих событий достаточно просто перехватывать эти системные вызовы и журнализировать информацию об этих событиях в соответствующих системных журналах, каждый из которых включает экземпляр файла с данными за последние сутки, а также - несколько (скажем, N) экземпляров за N предпоследних суток, обновляемых в режиме ротации. “Место” хранения этих файлов задаётся системным администратором в конфигурационном файле `/etc/audit/auditd.conf`. Состав системных журналов определяется составом правил отслеживания системной информации, которые изначально конфигурируются системным администратором в файле `/etc/audit/audit.rules`. *Содержимое этого файла может быть оперативно модифицировано без необходимости перезагрузки ОС* с использованием команды (утилиты) `auditctl`, позволяющей, в частности, *выдать список действующих правил, добавить новое правило, удалить одно или все заданные в конфигурационном файле правила*.

Подсистема аудита ОС Linux включает в свой состав демон `auditd` и следующие вспомогательные утилиты:

- `auditctl` - обеспечивает управление процессом аудита; позволяет получать информацию о текущем состоянии подсистемы аудита, а также добавлять и удалять правила
- `autrace` - обеспечивает отслеживание событий прикладных процессов, связанных с выполнением системных вызовов и сигналов ядру ОС
- `ausearch` - обеспечивает поиск информации о различных событиях в системных журналах
- `aureport` - выполняет генерацию различных отчётов о работе системы аудита.

Рассмотренные средства системного аудита не обеспечивают возможности аудита уровня С2 – например, восстановить значения изменённых записей файлов можно только из их резервных копий, созданных между моментами последнего и предпоследнего изменения этих записей. Но отметим, что реализация систем аудита уровня С2 либо основывается на какой-то из систем рассмотренного типа, либо реализует основную функциональность таких систем в своём составе.

Средства аудита различных приложений

Многие приложения, такие например, как почтовые серверы, веб-серверы, серверы баз данных и ряд других приложений имеют свои собственные средства аудита, предоставляющие возможность журнализировать в своем собственном логе для последующего анализа сведения о событиях, специфичных для каждого из приложений. Так, например, для почтового сервера, использующего журнал /var/log/maillog, к числу таких событий относятся успешные и безуспешные попытки отправки почтовых сообщений их адресатам, факты успешного приёма писем, или их возврата отправителю, например в связи с отсутствием на сервере почтового ящика указанного в письме получателя и пр.

Совершенно очевидно, что на сервере любого другого типа обязательно происходят какие-то события столь же специфичные именно для этого типа серверов. Поэтому необходимость наличия средств аудита “заточенных” на учёт событий, специфичных для различных типов серверов, совершенно очевидна. В своей реализации такие средства для некоторых приложений могут быть встроенными в реализацию этих приложений. Для других приложений средства аудита могут быть оформлены в виде отдельных программных продуктов. Так, например, для серверов баз данных средства аудита уровня

С2 изначально создавались как отдельные весьма не дешёвые программные продукты. Но постепенно производители серверов баз данных стали включать реализацию средств аудита в реализацию своего основного продукта.

Рассмотрим 2 подкласса обсуждаемого класса средств аудита: средства аудита относительно простых приложений и средства аудита систем (односерверных и многосерверных) баз данных.

Средства аудита относительно простых приложений

Относительно простыми приложениями являются, например, уже упомянутые выше сервер электронной почты и веб сервер. Системы аудита этих приложений встроены в реализацию самих приложений и имеют простейшую организацию Для журнализации событий, специфичных для приложения, в каждом из таких приложений используется единственный системный журнал. Так информация о событиях почтового сервера в системах UNIX обычно регистрируется в журнале `/var/log/maillog`, а о событиях веб-сервера, работающего под управлением демона Apache, - в журнале `/var/log/apache`. Никаких стандартных средств анализа и/или избирательной выдачи информации из этих журналов как правило не предоставляется. Предполагается, что указанные средства могут быть относительно легко созданы по принципу “сделай сам” и, возможно потом распространены внутри некоторого сообщества администраторов указанных приложений. Рассмотренные средства не обеспечивают аудита уровня С2. Так что, например, для восстановления контента веб-сайта после выполнения некоторых недопустимых изменений этого контента можно воспользоваться лишь его восстановлением из некоторой “достаточно свежей” резервной копии.

Средства аудита систем баз данных

Средства указанного типа включают в свой состав достаточно широкий спектр систем аудита, ориентированных как на односерверные конфигурации контролируемых серверов баз данных, так и на многосерверные гетерогенные конфигурации.

Средства аудита односерверных БД

В настоящее время многие производители серверов БД встраивают в реализацию своих серверов развитые средства аудита вплоть до уровня С2, предоставляя, однако, пользователям своих серверов некоторые возможности конфигурирования “включаемого в работу” уровня средств аудита (например, стандартного или С2). Это связано с тем, что повышение уровня применяемых средств аудита неизбежно влечёт как существенное

повышение нагрузки на процессор, так и ещё более существенное увеличение объёма внешней памяти, необходимой для хранения системных журналов. Если политика ИБ организации, использующий конкретный сервер БД такова, что этой организации достаточно уровня ИБ, обеспечиваемого стандартными средствами аудита, но неприемлемы затраты на приобретение более производительных компьютеров и более ёмких систем хранения данных, то при конфигурировании сервера БД достаточно задать стандартный уровень аудита. Не останавливаясь подробнее на затронутых вопросах отметим лишь, что, например, при конфигурировании сервера БД MS SQL Server по умолчанию выбирается стандартный уровень аудита. Для включения режима аудита уровня C2 необходимо задать параметр конфигурации "c2 audit mode".

Средства аудита многосерверных гетерогенных конфигураций БД

Интеграция локальных систем аудита многосерверных гетерогенных конфигураций БД может быть выполнена лишь специальными системами аудита, реализованными в качестве отдельного программного продукта. Наиболее известным программным продуктом такого типа является Oracle Audit Vault. Эта система позволяет собирать и консолидировать воедино данные аудита, получаемые из нескольких источников, хранить их в безопасном хранилище данных аудита, обнаруживать события, удовлетворяющие интересующим аудитора свойствам и, естественно генерировать разнообразные отчеты. Система Oracle Audit Vault поддерживает сбор данных аудита, генерируемых: серверами

- Oracle Database 9 i, выпуск 2 (9.2)
- Oracle Database 10 g выпуск 2 (10.2)
- Oracle Database 11 g выпуск 1 (11.1)
- Microsoft SQL Server 2000
- Microsoft SQL Server 2005

При этом данные аудита могут собираться из таблиц журнала аудита БД Oracle, файлов аудита операционной системы серверов БД Oracle, системных журналов Windows, журналов аудита C2 для баз данных MS SQL Server и ряда других источников. Более подробное рассмотрение этой системы выходит за рамки настоящего учебника. Интересующиеся могут найти много информации о системе через поисковые серверы интернета, введя в поисковой строке название этой системы.

О комплексных системах аудита ИТ-инфраструктур КСС

Системы рассматриваемого здесь класса обеспечивают комплексный (интегрированный) аудит КСС, включающих в свой состав серверы различных сетевых служб и использующих для организации аудита всей КСС в целом данные:
- подсистем аудита серверов всех этих служб,
- операционных систем входящих в состав КСС компьютеров
- и, возможно, различную дополнительную информацию.

При этом мы ограничимся очень краткой характеристикой широко известной современной комплексной системы аудита - Netwrix Auditor.

Система Netwrix Auditor поддерживает C2 аудит для КСС, построенных с использованием ИТ-систем очень широкого и разнообразного спектра, включая:
- MS Active Directory,
- Azure AD (Active Directory),
- MS Exchange,
- MS SharePoint,
- MS SQL Server,
- серверы базы данных Oracle,
- системы виртуализации VMware,
- системы хранения данных EMC и NetApp,
- MS Windows Server,
- клиентские ОС MS Windows,
- MS Office 365 и ряд других систем и продуктов.

В дополнение к информации о событиях, журналируемых системами аудита всех используемых в составе КСС, анализируемой средствами входящих в Netwrix Auditor программных и программно-аппаратных средств, система Netwrix Auditor предоставляет возможность регистрации весьма разнообразной дополнительной информации. Эта информация может быть использована для последующего избирательного поиска и генерации широкого круга избирательных отчётов, ориентированных на различные категории “читателей” этих отчётов.

В качестве одного из типов такой дополнительной информации следует упомянуть возможность использования видеозаписи действий пользователей, включая видеозапись происходящего на мониторе компьютера пользователя. В последнем случае обеспечивается возможность записи не только происходящего на экране монитора, но и

метаданных, связанных с представленными на экране объектами, таких, например, как: заголовки окон, названий обслуживающих эти окна процессов и пр. И, естественно, система предоставляет средства поиска интересующих аудитора событий, зарегистрированных рассмотренным способом.

Естественно, наличие рассмотренных средств, крайне затруднит работу взломщиков-инсайдеров, пытающихся “пробить” систему ИБ КСС изнутри.

Охарактеризовав систему Netwrix Auditor буквально “несколькими штрихами”, мы прекращаем её дальнейшее рассмотрение, ибо это - задача отдельной книги, а не курса по сетевым технологиям. Читатель же, заинтересовавшийся этой системой сможет найти массу информации об этой системе через поисковые системы интернета. А для иллюстрации степени массовости публикаций в интернете, посвящённых этой системе, сообщим, что, например, по поисковому запрос с текстом “C2 audit” будет выдано несравненно больше ссылок на различные описания системы Netwrix Auditor, чем на собственно описание сути C2 аудита. А по поисковому запросу «C2 аудит» в по крайней мере нескольких первых страницах ссылок на станицы, описывающие суть C2 аудита вообще не будет.