Лекция 14
Вредоносные программные воздействия. Меры по защите локальных ИС

Предметом основного рассмотрения первой части настоящей лекции являются различные программные воздействия на КСС, нарушающие её ИБ в направлении некорректного поведения системы программного обеспечения КСС, которая в результате этих воздействий либо перестаёт делать что-либо из того, что она должна делать, либо начинает делать совершенно непредусмотренные вещи. Такие программные воздействия называются вредоносными. В настоящем параграфе проводится классификация таких воздействий, рассматриваются возможные виды производимого ими разрушительного эффекта, детально рассматриваются особенности и способы реализации, а также рассматривается специфика каждого из видов вредоносных воздействий, методы их “проникновения” на компьютер и вкратце рассматриваются меры по борьбе с ними.

14.1. Общая характеристика вредоносных программных воздействий

Рассмотрим основные виды вредоносных программных воздействий. К ним относятся: вредоносные программы, взломы внутренней системы защиты (внутренние атаки) и взломы сетевой системы защиты (сетевые атаки). Рассмотрению каждого из этих видов (кроме последнего) посвящён отдельный подраздел настоящей лекции. Рассмотрению внешних атак посвящена следующая лекция.

Здесь же мы вкратце рассмотрим, как указанные виды вредоносных программных воздействий могут комбинироваться друг с другом и в чём может состоять “разрушительный” эффект всех этих воздействий.

Вначале отметим, что вредоносные программы по их способу проникновения в компьютер разделяются на вирусы, троянские программы и сетевые черви. Подробное рассмотрение этих типов вредоносных программ будет проведено чуть позже . А для понимания смысла двух следующих утверждений достаточно интуитивного представления об этих типах программ. Во-первых, *сетевые атаки для выполнения действий по проникновению на атакованный компьютер могут использовать методы распространения компьютерных червей*. А во-вторых, *атакующий программный код, внедряемый на атакуемый компьютер при совершении сетевой атаки на него, может быть “оформлен” в виде вируса* и,

кроме выполнения собственного атакующего воздействия, может продолжить своё дальнейшее распространение на другие компьютеры методами, используемыми для этого компьютерными вирусами.

Перейдем к рассмотрению возможных видов разрушительного (по отношению к ИБ КСС) эффекта, который могут производить вредоносные программные воздействия всех типов после “появления” вредоносного кода на некотором компьютере. В число этих действий могут входить:

- ***передача по сети*** на компьютер злоумышленника **конфиденциальной информации** (логинов и паролей, секретных файлов, информации о нажатии клавиш клавиатуры и пр.);
- ***блокировка компьютера*** и выдача на экран сообщения с требованием выкупа;
- ***искажение или удаление файлов*** данных и программных файлов вплоть до переформатирования жёсткого диска;
- ***похищение денег*** при внедрении в одно из приложений, оперирующих с финансами;
- ***физическое повреждение жёстких дисков***; это может быть сделано путём перемещения считывающих головок с частотой, резонансной собственной частоте колебаний оси дисковода; современные дисководы аппаратно предотвращают возможность сделать это;
- ***физическое повреждение процессора***; может быть выполнено путём циклического выполнения команд процессора, выполнение которых сопровождается повышенным тепловыделением; в результате могут оплавляться зоны кристалла процессора, реализующие соответствующие команды; в современных процессорах такая ситуация невозможна;
- ряд других не менее “приятных” эффектов.

Все эти возможные эффекты должны быть по возможности учтены при выработке политики ИБ.

В завершение настоящего пункта укажем, что разработка вредоносных программ и средств реализации внутренних и сетевых атак требует высокой квалификации специалистов, выполняющих такую разработку, и способности “врубаться” в разнообразные тонкости процессов выполнения и взаимодействия программ, функционирующих как в локальной, так и в сетевой среде. Поэтому таких специалистов называют хакерами (от глагола “to hack“ - делать зарубку). Интересно отметить, что *на заре программирования слово “хакер” имело совсем*

другой, восторженно-положительный, а не криминальный смысл. Хакерами тогда называли программистов, которые могли быстро и элегантно устранять сложно обнаруживаемые ошибки в программах, быстро “врубаясь” как тончайшие нюансы поведения этих программ.

14.2. Вредоносные программы

Наиболее массово встречающейся широкому кругу пользователей разновидностью *вредоносных программных воздействий* являются *вредоносные программы*, которые по способу их проникновения в компьютер разделяются на компьютерные вирусы, троянские программы (или просто “трояны”) и сетевые черви.

Вирусы - это вредоносные программы (фрагменты программного кода), которые:

- встраиваются в программы (исполнимый код), документы (посредством макросов), электронные письма, веб-страницы и пр. (“заражают” указанные объекты);
- саморазмножаются, копируя себя и внедряя свои копии в другие найденные программы, документы и отправляемые вирусом электронные письма; указанные копии запускаются снова при выполнении программ, открытии документов, открытии электронных писем;
- переносятся с объектом, который они заразили при выполнении операций копирования файлов, отправке электронных писем и т.д.;
- маскируются с использованием множества способов маскировки.

Вирусы могут быть классифицированы по среде “обитания”, по особенностям работы и по степени разрушительности.

По среде обитания основными видами вирусов являются:

- *файловые* - они внедряются в файлы исполнимых программ; отметим, что Postscript-файлы - это тоже исполнимые программы, поскольку язык Postscript является хотя и специализированным, но полнофункциональным языком программирования;
- *загрузочные* - встраиваются в загрузочный сектор носителя информации (постоянного или съёмного), из которого возможна загрузка ОС, запускаются при загрузке ОС;
- *макровирусы* - заражают любые документы приложений из MS Office (Word, Excell, Powerpoint, Access), поскольку эти приложения поддерживают применение макросов при создании документов;
- *скриптовые* - встраиваются в HTML-страницы сайтов и запускаются при открытии заражённых страниц;

- *почтовые* - передаются в электронных письмах и срабатывают при их открытии.

По особенностям работы следует отметить следующие виды сильно “заразных” и с трудом поддающихся “лечению” вирусов:

- *резидентные* - эти вирусы после однократного запуска зараженной ими программы “оседают” в оперативной памяти компьютера и продолжают свою “инфецирующую” деятельность после завершения работы этой программы;
- *вирусы-невидимки* или стелс-вирусы (stealth) - разновидность резидентных вирусов; вирусы-невидимки фальсифицируют читаемую с диска информацию, так, что программа, которой предназначена эта информация, получает неверные данные; технология построения таких вирусов, называемая Стэлс-технологией, может использоваться при создании вирусов с различной средой обитания: загрузочных, файловых и даже макровирусов.
- *полиморфные или самошифрующиеся* - вирусы, предпринимающие специальные меры для затруднения своего обнаружения и анализа; обычно 2 любых экземпляра такого вируса не содержат ни одного постоянного участка кода (по которым обычно распознаётся вирус); это достигается путём шифрования основного тела вируса и существенной модификации от копии к копии модуля-расшифровщика.

По степени разрушительности

Различные вирусы любых типов могут иметь разную степень разрушительности и могут быть как совершенно безвредными, так и крайне разрушительными.

Основные способы проникновения вирусов в компьютер в подавляющем большинстве случаев так или иначе связаны с использованием программ и данных, полученных из *ненадёжных источников*, к числу которых можно отнести:

- *съёмные носители информации* устанавливаемые в компьютер; это могут быть как носители других пользователей, так и собственные носители, которые перед этим устанавливались на стороннем компьютере;
- *пиратские копии программ и документов* (как правило со взломанной защитой от копирования), полученные из различных источников (сайтов, приятелей, розничного рынка электронных товаров и пр.)

- *лже-копии известных сайтов* (например, сайтов производителей программного обеспечения), содержащие зараженные вирусами (и, возможно, троянскими программами) копии известных программ; страницы таких сайтов обычно имеют вид максимально близкий к виду “копируемого” файла, но заведомо отличаются адресной строкой, отображаемой в браузере;
- *неизвестные сайты высоко привлекательного содержания*; неизвестность таких сайтов состоит в том, что по их доменным именам практически невозможно определить, кому принадлежат эти сайты; высокая привлекательность указанных сайтов для широких кругов пользователей может состоять, например, в том, что их содержанием являются “сногсшибательные” новости, “волшебные” рецепты от всех болезней, информация фривольного содержания и пр.; а опасность таких сайтов состоит в том, что их станицы могут быть “нашпигованы” скриптовыми вирусами, которые могут “сработать” при открытии этих страниц;
- *электронные письма от неизвестных отправителей*; такие письма могут быть не просто спамом, а быть зараженными почтовыми вирусами (ими заражаются сами почтовые сообщения), так и включать заражённые различными файловыми вирусами вложения

Рассмотренный список ненадёжных источников информации может быть расширен.

Перейдем к ***мерам по борьбе с вирусами***. В их число входят организационные меры (инструкции, регламенты, контроль за исполнением запрета использования ненадёжных источников информации), программно технические меры, состоящие в надлежащем использовании и регулярном обновлении хороших антивирусных программ и выполнение регулярного аудита аномального поведения системы.

Критерием качества антивирусных программ является способность обнаружения и нейтрализации (“лечения”) не только простых вирусов, достаточно легко обнаруживаемых по их сигнатурам (шаблонам входящих в вирусы кодовых фрагментов), но и сложно организованных вирусов, таких как рассмотренные выше резидентные вирусы, вирусы-невидимки и самошифрующиеся вирусы. Известным примером хороших антивирусных программ, на наш взгляд, являются антивирусы Касперского и DrWeb.

Необходимость регулярного обновления версий и модификаций антивирусных программ обусловлена тем, что в своей работе такие программы используют различные базы данных об известных вирусах, обязательно включающих в свой состав базу сигнатур известных вирусов. Поиск вирусов осуществляется на основе использования информации из баз данных об известных вирусах. Поэтому такие программы могут обнаружить и “излечить” лишь известные вирусы. Для того, чтобы антивирусная программа смогла обнаружить ранее неизвестный ей вирус, она должна использовать обновлённую БД об известных вирусах. Такие обновлённые БД обычно не поставляются отдельно, а входят в новые версии и модификации антивирусных программ или во временные программы-заплатки (патчи - patch).

Но каким образом обнаруживается сам факт заражения компьютера? Дело в том, что вирус не всегда явно и “громко” сразу же проявляет себя, а может вести какую-либо малозаметную деятельность (например, регулярную отправку относительно небольших порций информации через сеть). Но если суметь обнаружить факт внезапного начала таких отправок, можно прийти к выводу о достаточно высокой вероятности, того, что произошло какое-то вредоносное программное воздействие на компьютер. Это может быть вредоносное воздействие любого из упоминавшихся выше типов. Для выяснения того, что в действительности произошло нужен дополнительный анализ. Выявление аномального поведения компьютеров и выяснение причин обнаруженных аномалий может выполняться методами аудита, средства проведения которого вкратце рассматриваются в лекции 16.

Троянские программы (называемые также троянами, программами с закладками или шпионскими программами), также, как и известный Троянский конь, внутри являются совсем не такими, как кажутся снаружи, скрывая под безобидной внешностью опасную начинку. Более детально, троянские программы характеризуются тем, что они

- *маскируются* под известные полезные программы, но *содержат скрываемую от пользователя функциональность*, нарушающую ИБ компьютера;
- *не размножаются*;
- обладают средствами *активного сопротивления удалению с компьютера*;

- *могут быть скопированы, отправлены по электронной почте и пр.* как обычные программные файлы.

Подобно вирусам троянские программы (закладки) могут распространяться путём получения их из различных ненадёжных источников информации, указанных при рассмотрении вирусов. Однако следует отметить, что организации, работающие с особо конфиденциальной информацией, в частности, с информацией составляющей государственную и/или коммерческую тайну, рассматривают как потенциально содержащие шпионские закладки все программы, поставляемые им производителями этих программ лишь в бинарном виде, без исходных текстов. А для программ, поставляемых вместе с исходными текстами, таким организациям требуется наличие для этих программ сертификатов ИБ (гарантирующих отсутствие в программах шпионских закладок), выданных ФСБ (Федеральной службой безопасности) и/или ФСТЭК (Федеральной службой по таможенному и экспортному контролю).

Вредоносный эффект троянских программ также может быть самым разным: от практически безобидного до крайне опасного. Отметим лишь, что одним из наиболее распространённых видов вредоносной деятельности троянов является однократная или регулярная отправка различной конфиденциальной информации злоумышленнику, заказавшему их изготовление. Именно поэтому одним из названий таких программ является название “шпионские программы”.

Меры борьбы с троянскими программами также совпадают с мерами борьбы с вирусами. При этом надо отметить, что если используемые антивирусные программы не являются одновременно ещё и антишпионскими (AntiSpy), то дополнительно к антивирусным программам должны применяться и регулярно обновляться какие-либо антишпионские программы. Что касается упоминавшегося выше антивируса Касперского, то он обеспечивает обнаружение и нейтрализацию как вирусов, так и троянских программ.

Сетевые черви (worms) исторически являются первым видом вредоносных программ, созданных ещё до появления персональных компьютеров, являющихся основной “средой обитания” вирусов и троянов. Сетевые черви это программы, которые:

- *проникают на компьютер,* пользуясь ошибками и уязвимостями программ, работающих на компьютере и принимающих данные из сети (об этих уязвимости говорится далее при рассмотрении сетевых атак);

- *саморазмножаются*, проникая с заражённого компьютера на найденные в сети ещё незаражённые компьютеры (но не заражают другие файлы на данном компьютере);
- *оказывают вредоносный эффект* на текущем компьютере;
- может вызывать *неустойчивую работу компьютеров сети* (включая как зараженные, так и ещё не зараженные) из-за особенностей проникновения сетевых червей.

Очевидно, что также как для вирусов и троянов , вредоносный эффект, оказываемый сетевыми червями может быть самым разным как по степени его опасности, так и виду производимого разрушительного действия.

Методы борьбы с сетевыми червями включают как применение обеспечивающих их поиск на компьютере антивирусных программ, так и методы, применяемые для борьбы с сетевыми атаками (в частности - межсетевые экраны), рассматриваемые в лекции 16.

14.3. Внутренние атаки

Внутренней атакой называется *взлом системы ИБ* компьютера, достигаемый путём несанкционированного получения *взломщиком* (являющимся одним из зарегистрированных в системе пользователей, и, поэтому, иногда называемому *инсайдером*) полномочий и конфиденциальной информации какого-либо высоко привилегированного пользователя системы (например, суперпользователя root). Этой информацией инсайдер может воспользоваться как самостоятельно (для выполнения требуемых ему действий, нарушающих систему ИБ КСС), так и для передачи некоторому внешнему лицу для непосредственного использования им конфиденциальной информации (например, документов с описанием содержания секретных разработок) или для последующего выполнения этим лицом внешних атак на КСС. Отметим также, что указанная информация иногда может быть получена внешним лицом и без привлечения инсайдера, например:

- путём успешного выполнения внешнего наблюдения за терминалами (экранами мониторов и клавиатурой) и печатающими устройствами

а также путём внедрения в КСС через вирусы или троянские программы шпионских программ, собирающих и передающих ему по сети требуемую информацию.

Основные виды внутренних атак

Доступ к информации о логине и пароле высоко привилегированного пользователя является наиболее простым для понимания способом получения высоких полномочий. К основным способам доступа к такой информации относятся визуальное наблюдение, а также поиск доступных на чтение копий файлов паролей пользователей и выполнение взлома найденных файлов.

Остановимся подробнее на 2-м способе. Файлы паролей (в различных разновидностях системы UNIX это файлы /etc/master.passwd или /etc/shadow), в соответствии с правильно установленными правами доступа к этим файлам, доступны только суперпользователю (или программам, выполняющимся от его имени). Но суперпользователь случайно может сохранить копию файла паролей с открытыми правами на чтение для всех пользователей или пользователей некоторой группы, к которой принадлежит и сам суперпользователь. Поскольку в указанном файле все пароли хранятся в зашифрованном виде и, поскольку используемый для этого шифрования алгоритм не является обратимым (по зашифрованному паролю невозможно восстановить исходный пароль), на первый взгляд кажется, что наличие доступной копии зашифрованных паролей не несёт никакой угрозы. Но так кажется только на первый взгляд. Используя такой файл пароли могут быть взломаны методом грубой силы. Суть этого метода чрезвычайно проста: последовательно генерируются все цепочки символов, начиная с цепочек единичной длины с последующим полным перебором всех цепочек всё бóльшей и бóльшей длины. В процессе перебора каждая из сгенерированных цепочек символов шифруется методом, применяемым для шифрования паролей, и сравнивается с полем зашифрованного пароля всех записей файла паролей. Если для некоторой строки файла произошло совпадение её значения с зашифрованной текущей цепочкой, то эта текущая цепочка является паролем пользователя, логин которого указан в той же строке файла.

Интуитивно ясно, что полный перебор цепочек символов с шифрованием каждой из перебираемых цепочек требует больших вычислительных ресурсов, вопрос в том лишь, в том, насколько больших. Простым ответом на этот вопрос является утверждение о том, что для взлома этим методом пароля из не более, чем 5-ти алфавитно-цифровых символов (при использовании одного регистра латинского алфавита) с использованием для этого взлома современного персонального компьютера не потребуется и получаса. Это время может быть

существенно уменьшено путём предварительного перебора всех словарных слов (их всего лишь примерно 50 тысяч) или дат рождения (из всего лишь чуть более 36500 за 100 лет), поскольку многие неопытные пользователи любят применять в качестве пользователей именно осмысленные слова и даты.

В общем случае вычислительная сложность взлома пароля равна La^{Lp}, где *La* – это длина алфавита, используемого при написании пароля, а *Lp* – длина пароля в символах. Длина (или мощность множества) алфавита может быть увеличена по сравнению с 36 (длиной алфавита, упоминаемого выше) не менее, чем в 5 раз за счёт использования не только латинских, но и кириллических букв, вводимых на разных регистрах. При таком увеличении длины алфавита вычислительная сложность и время взлома пароля длиною 5 символов возрастают в 5^5 = 3125 раз, а каждое увеличение длины пароля на 1 символ влечёт увеличение времени его взлома в *La* (больше чем в 36*5=180) раз.

После проведенного рассмотрения становится совершенно очевидным что при выборе пароля для повышения его стойкости от взлома необходимо максимально расширять алфавит используемых символов и повышать длину пароля. Теперь читателям станут совершенно понятными требования, предъявляемые к паролям многими известными сетевыми службами: обязательное использование в пароле длиной не менее 8 символов букв на разных регистрах, цифр и специальных символов. Если при этом читатель будет использовать буквы как латинского, так и кириллического (или другого национального) алфавита - он тем самым ещё больше повысит стойкость пароля к потенциальным попыткам его взлома.

Отметим, что для инсайдера КСС возможные копии файлов паролей являются не единственными источниками паролей других зарегистрированных пользователей КСС. Такой инсайдер может установить на своём компьютере специальную программу, называемую *снифферо*м (sniffer - вынюхиватель), выполняющую прослушивание всего трафика, проходящего через сегмент сети, к которому подключён компьютер инсайдера, и “извлечение” из этого файла всех пар логинов и паролей, пересылаемых через этот сегмент при установлении соединений с различными сетевыми службами. При этом изощрённый сниффер может перехватывать не только логины и пароли, передаваемые в открытом виде при установлении сетевых соединений соответствующими службами (FTP, TELNET и др.), но и логины с зашифрованными стандартным образом для систем UNIX паролями, пересылаемыми при установлении соединений между

компьютерами для служб, использующих возможности автологина (RLOGIN, RSH, RCP). В последнем случае пароль также может быть взломан рассмотренным выше методом грубой силы, позволяющим по зашифрованному паролю подобрать соответствующий ему символьный пароль. Правда, дополнительно потребуется подобрать логин к этому паролю, но для этого можно “перебрать” все логины из файла /etc/passwd и экспериментально проверить, подходит ли для какого-то из логинов взломанный пароль.

Получение прав доступа привилегированного пользователя даже при кратковременном доступе к его терминалу

Доступ не уполномоченных на то лиц к терминалу с открытым интерфейсом одного из высоко привилегированных пользователей системы может быть использован злоумышленником для быстрого получения им полномочий владельца терминала. Рассмотрим, например, какие действия, обеспечивающие дальнейший доступ злоумышленника к системе с правами суперпользователя `root`, может выполнить этот злоумышленник, если получит всего лишь 30-секундный бесконтрольный доступ к символьному терминалу (либо графическому терминалу с открытым терминальным окном), находящемуся в состоянии открытого сеанса работы суперпользователя **`root`**. Эти действия состоят в выполнении 2-х коротких команд:

```
cp /bin/sh /home/my-login/sh
chmod 4555 /home/my-login/sh
```

При этом в качестве имени каталога своего домашнего **`my-login`** злоумышленник должен указать своё логин-имя в системе.

При помощи первой из рассмотренных команд выполняется копирование командного интерпретатора **`sh`** в домашний каталог злоумышленника. Отметим что владельцем копии этого файла останется **`root.`** Вторая команда изменяет права доступа к копии **`sh`**, устанавливая для неё бит SetUID и предоставляя права на чтение и выполнение этого файла всем пользователям системы. А установленный бит SetUID приведёт к тому, что при запуске на выполнение копии `sh` она будет выполняться с правами владельца этой копии, т.е. `root`; с этими же правами будет выполняться каждая запущенная этой копией `sh` команда. В итоге злоумышленник в удобное для себя время сможет превратить свой терминал в терминал суперпользователя и получить все права работы в системе.

Наличие в системе командных SetUD файлов суперпользователя, доступных для выполнения другим пользователям

В системе не должно быть с установленным битом SetUID. При запуске такого файла для его выполнения текущим экземпляром командного интерпретатора выполняется вызов нового экземпляра командного интерпретатора. Этот интерпретатор запускается с правами суперпользователя root, поскольку он является владельцем интерпретируемого командного файла и поскольку для этого файла установлен бит SetUID. А, кроме того, существуют приёмы такого искусственного прерывания выполнения файлов (намеренно не будем их описывать), при котором файл завершается без восстановления действующих на момент вызова порождённого интерпретатора прав доступа пользователя, вызвавшего командный файл. В результате после такого прерывания пользователь продолжит работу в текущем экземпляре командного интерпретатора, но с правами суперпользователя root. И, естественно, что этот пользователь сможет сделать в системе всё, что ему заблагорассудится.

Отметим, что наличие бита SetUD в правах доступа к не предусмотренным в инсталляции системы двоичным исполнимым файлам суперпользователя тоже является признаком потенциальной опасности. Эти файлы потенциально могут обладать брешами в системе защиты (см. ниже), проникнув через которые злоумышленник сразу получить права доступа суперпользователя.

Очевидно, что наличие в системе файлов рассмотренного типа может быть связано либо с незнанием опасности этого суперпользователем, либо его ошибкой/небрежностью, либо с результатами какого-то (другого) взлома системы ИБ. В данный момент наша задача – избавить будущих суперпользователей от опасного незнания. На мерах борьбы с другими случаями мы остановимся в конце лекции.

Внедрение вредоносного кода в программу, запущенную привилегированным пользователем, через брешь в этой программе

Брешью или *уязвимостью* в системе защиты программы называется способ входа в программу, выполняемый **без проверки полномочий пользователя**. Иногда такой способ входа возможен через созданную при разработке программы и не удалённую после завершения её отладки отладочную точку входа. Но чаще способы такого входа базируются на обнаружении и злонамеренном использовании отсутствия тех или иных проверок в программе.

В качестве примера уязвимости последнего типа приведём обнаруженную одной из первых уязвимость системного вызова **gets()**, обеспечивающего ввод строки символов, завершающейся символом перевода строки, из стандартного входного потока программы. В качестве источника такого потока могут выступать клавиатура терминала, некоторый файл или стандартный выходной поток программы, предшествующей программе, содержащей вызов **gets,** в некоторой конвейерной конструкции. Уязвимость вызова **gets** состоит в том, что единственным параметром этого вызова является адрес буфера, предназначенного для ввода строки. И, если при вводе этой строки количество её символов, предшествующих переводу строки, больше длины указанного буфера, то произойдёт переполнение этого буфера и замена в памяти значений последовательности байтов, следующих за этим буфером, на значения символов, переполнивших буфер.

Эта уязвимость была использована при изобретении следующего способа внедрения вредоносного кода в программу, использующую системный вызов **gets**, схема реализации которого приведена на рис. 14.1.

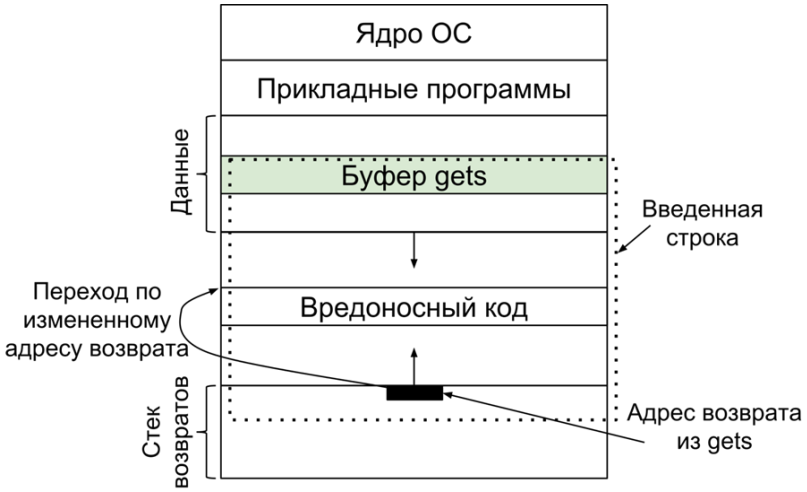

Рис. 14.1. Схема внедрения вредоносного кода с использованием уязвимости `gets`

Основу этого рисунка составляют схема распределения памяти компьютера и наложенная на него схема запуска на этом компьютере вредоносного кода, содержащегося внутри введенной при помощи **gets** строки символов. При стандартном распределении памяти компьютера в начале этой памяти располагается ядро ОС, затем коды прикладных программ, вслед за которыми размещается область данных. При использовании прикладных программ, полученных путём компиляции программ с языков программирования с блочной структурой область данных растёт вниз (в направлении б'ольших адресов) при

входах во вложенные блоки и при вызове процедур (функций). В нижней части памяти расположен *стек возвратов*, каждый элемент которого содержит значения регистров компьютера в момент вызова процедуры (функции), *адрес возврата* (для передачи управления после завершения выполнения процедуры) и некоторую дополнительную информацию.

Для выполнения внедрения вредоносного кода на вход программы, вызывающей `gets`, подаётся длинная строка, выходящая за пределы выделенного для её ввода буфера, перекрывающая всю свободную область памяти и начальную часть стека возвратов. При этом в поле адреса возврата верхнего элемента стека заносится адрес начала содержащегося во введённой строке вредоносного кода. В результате сразу после завершения ввода строки управление будет передано прямо на только что внедрённый вредоносный код. А он, естественно, сможет выполнить любые действия, которые правомочен выполнить пользователь, вызвавший содержащую вызов программы. Если эту программу вызвал суперпользователь, то, в частности, можно программно выполнить и действия по сохранению в каталоге взломщика копии командного интерпретатора с установленным битом SetUID. Это позволит взломщику оставить себе время для выбора момента начала требуемого ему вредоносного воздействия и тщательной проработки содержания этого воздействия

Нами рассмотрена лишь общая схема осуществления вторжения и не рассматриваются детали, связанные с методом определения адреса верхнего элемента стека возвратов, необходимого для подмены поля адреса возврата. Это не входит в задачи нашего курса. Отметим лишь, что проблема подачи на вход взламываемой программы строки любой длины и содержащей любые символы (включая не вводимые с клавиатуры) легко решается путём подачи такой строки из файла или из стандартного выходного потока другой программы.

Естественно, что способ устранения этой уязвимости был быстро найден. Он состоит в замене всех системных вызовов **`gets`**, не контролирующих переполнение буфера ввода, на аналогичную функцию, но вводящую в указанный буфер не больше L первых символов строки, где L - длина буфера, задаваемая вторым параметром функции. Естественно, что соответствующей переработке подверглись тексты всех известных программ. Но, возможно где-то сохранились старые версии каких-либо программ. Поэтому во избежание возможного взлома системы ИБ рассмотренным или каким-то более новым способом категорически рекомендуется использовать последние версии всех программ а также - устанавливать так называемые патчи (patch - заплатка), временные программы-заплатки оперативно создаваемые производителями программ, для которых найден новый способ внедрения в них вредоносного кода. Установка таких патчей устраняет обнаруженную уязвимость.

Отметим, что после обнаружения рассмотренной уязвимости был найден ряд других брешей, основанных на переполнении буфера, и, естественно, были найдены способы их устранения. Но существуют уязвимости и других типов. Обнаружить их можно с использованием специальных программ - сканеров безопасности, рассматриваемых в лекции 16.

14.4. Меры по борьбе с внутренними атаками и их последствиями

Перед рассмотрением *технических мер* отметим чрезвычайную важность мер организационных. Ограничение доступа в помещение, в котором работают привилегированные пользователи, а также неукоснительное выполнение должностных инструкций начиная с простых правил доступа к терминалу и выбора стойких паролей и завершая выполнением комплекса мер, рассматриваемых ниже.

Сразу отметим, что ни какая-либо одна из рассматриваемых ниже мер, ни совместное применение всего комплекса этих мер не может гарантировать “абсолютную непробиваемость” системы защиты от внутренних атак. Однако комплексное применение всех рассмотренных мер способно обеспечить как минимизацию вероятности успешной (для злоумышленника) атаки, так и её, по возможности, раннее обнаружение и возможную минимизацию её отрицательных последствий. Итак, рассмотрим эти меры.

Шифрование всей секретной и конфиденциальной информации

Всю секретную и конфиденциальную информацию следует надёжно шифровать методами, рассматриваемыми в лекции 17 во избежание потенциального доступа к ней злоумышленника, совершившего атакующие действия. При этом шифровать эту информацию должны её владельцы, а не суперпользователь, который с некоторой отличной от 0 вероятностью может быть законспирированным инсайдером, в задачи которого входит лишь получение и передача заказчику подобной информации. Если приложения, использующие такую информацию, получают её только из стандартного входного потока, а результаты обработки направляют только в стандартный выходной поток, то для дешифрации входных данных и шифрования выходных можно использовать различные криптографические утилиты (желательно включённые в конвейер с использующим их приложением). В противном случае программы приложений

должны сами выполнять шифрацию и дешифрацию, возможно с использованием соответствующих криптографических библиотек.

Регулярное обновление программного обеспечения

По мере эксплуатации любого программного обеспечения (ПО) в нём могут обнаруживаться ошибки и бреши. Это замечание касается не только прикладного и системного ПО, но и ПО, призванного бороться с вредоносными программами - антивирусного и антишпионского ПО. Для ПО первой из указанных категорий (прикладного и системного) в первую очередь характерно появление ошибок и брешей в самом ПО этой категории. А для антивирусных и антишпионских программ более характерно то, что с течением времени они оказываются не в состоянии обнаруживать и обезвреживать вновь появившиеся вредоносные программы (котором может “открыть ворота”) в систему недостаточно дисциплинированный пользователь этой системы.

Поэтому для обеспечения постоянно высокого уровня устойчивости КСС к возможным атакующим воздействиям необходимо постоянно обновлять всё используемое ПО. Следует отметить две основных разновидности такого обновления.

Первая из упомянутых разновидностей связана с оперативным исправлением вновь обнаруженных ошибок и брешей в системе защиты. Для исправления таких ошибок либо устранения брешей производителем ПО обычно оперативно выпускаются специальные программные “заплатки” (наскоро устраняющие обнаруженные изъяны), именуемые на программистском жаргоне патчами (patch - заплатка). В современных системах ПО как правило встроены средства автоматического обнаружения на сервере его производителя патчей, не установленных ещё для данного экземпляра системы ПО, а также автоматического скачивания обнаруженных патчей по сети и их выполнения. Выполнение каждого патча как раз и производит все изменения программы, необходимые для “латания” обнаруженного изъяна. Указанные действия выполняются на протяжении всего срока действия лицензии на используемое ПО. Пользователю КСС необходимо лишь не отказываться от скачивания и установки патчей. Для антивирусных и антишпионских программ их обновление при обнаружении новых разновидностей вредоносных программ как правило состоит в автоматически выполняемом (в течение срока действия лицензии на антивирусные/антишпионские программы) обновлении баз данных с сигнатурами

фрагментов вредоносного кода и иной информацией, позволяющими выполнять обнаружение и нейтрализацию всех вредоносных программ, информация о которых содержится в базе данных.

Отметим, однако, что существуют системы ПО (особенно среди небольших свободно распространяемых систем и программ), не реализующие рассмотренные выше механизмы автообновления. Появление новых брешей, скачивание и установку патчей, равно как появление новых версий систем и обновление этих версий в этом случае должно выполняться системным администратором “вручную”. Это накладывает на него достаточно большой объём дополнительной работы. Поэтому при выборе функционально идентичных разновидностей всех систем ПО, используемых в составе КСС, наличие у конкретной разновидности любой из указанных систем ПО средств автообновления должно рассматриваться как один из плюсов этой разновидности системы.

По мере накопления как объёма “запатченного” кода систем ПО, так и предложений по их функциональному развитию, периодически разрабатываются новые версии таких систем, которые рекомендуется внедрять вскоре после их появления. Дело в том, что старые версии таких систем со временем перестают поддерживаться их производителями. В частности, для таких старых версий перестают разрабатываться и обновляться патчи, “латающие” вновь обнаруженные бреши, и система ПО становится доступной для вторжения.

Регулярные проверки ПО КСС сканером безопасности

Несмотря на перманентное обновление всех используемых в КСС систем ПО существует вероятность, что в некоторых из этих систем всё ещё имеются какие-либо не устранённые бреши. Тремя основными причинами этого являются следующие. Во-первых, производитель какой-либо из систем ПО может изготовить патч некоторой бреши недостаточно оперативно по сравнению с другими производителями. Во-вторых, для систем, требующих ручного скачивания и установки патчей, в результате какого-либо организационного сбоя некоторые патчи могут оказаться не установленными. И, в-третьих, в составе ПО КСС могут применяться собственные и/или заказные системы ПО, созданные относительно небольшими производителями, которые не в состоянии в должной степени оперативно отслеживать появление новых типов уязвимостей и изготавливать соответствующие патчи. Поэтому регулярная проверка отсутствия уязвимостей

ПО систем КСС некоторыми дополнительными средствами никогда не будет излишней.

Наиболее известными из таких средств являются сканеры безопасности, рассматриваемые в лекции 16. Наиболее полную проверку наличия уязвимостей во всех программах позволяют выполнить локальные (host based) сканеры безопасности, запускаемые на самом проверяемом компьютере. В частности, они позволяют проанализировать код всех исполнимых программ и выявить в них все фрагменты кода, соответствующие тем или иным уязвимостям методами, аналогичными методам обнаружения вирусов и троянских программ. Другие аспекты применения сканеров безопасности будут рассмотрены нами при рассмотрении их применения при обнаружении уязвимостей программ, которые могут быть использованы злоумышленником при сетевом доступе к ним в рамках реализации сетевых атак.

Постоянный аудит состояния защищенности КСС

Отсутствие в ПО КСС всех известных уязвимостей вовсе не гарантирует от того, что в них не содержатся уязвимости, ещё не известные создателям патчей и сканеров безопасности. В результате использования таких уязвимостей и/или непосредственных действий возможного инсайдера какая-либо из систем ПО КСС может быть подвергнута атакующему воздействию, в результате которого функциональность некоторой системы ПО была “дополнена” некоторыми вредоносными элементами. Поскольку этот факт не обнаруживается ни одним из рассмотренных выше методов, выполненное атакующее воздействие оказывается успешным. Единственно возможным методом обнаружения такого факта успешной атаки такого типа является *периодический аудит* состояния системы защищенности КСС. Такой аудит включает анализ состояния и поведения КСС с целью обнаружения недопустимых изменений в них, а также анализ и определение возможных причин, приведших к таким изменениям.

Изменения в состоянии КСС может заключаться, например, в появлении в файловой системе компьютера неизвестных файлов, частности, файлов с установленным битом SetUID (особенно - командных файлов), потенциальная опасность которых рассмотрена нами выше. Обнаружение таких файлов может быть выполнено средствами стандартной в системах UNIX утилиты **`find`** поиска файлов по их атрибутам. Характеристики недопустимых изменений в БД отдельных систем ПО КСС и методы их обнаружения находятся скорее в области

интерпретации содержания БД, чем в области системного администрирования. Так, например, аномально большая сумма зарплаты того или иного сотрудника в БД сотрудников предприятия может оказаться результатом недопустимых изменений этой БД,

Недопустимый характер поведения КСС может проявляться, например, во внеурочном времени терминального входа в систему тех или иных пользователей (особенно - системного администратора), в фактах выполнения обычными пользователями команды su (переход в режим суперпользователя root), во внеурочном выполнении тех или иных действий, проявлении повышенной сетевой активности и пр. Все эти аномалии могут являться признаком выполненных злоумышленником атакующих действий. Значительная часть таких аномалий может быть обнаружена при помощи составляющих основу большинства систем аудита средств журнализации различных событий в системе. К ним относятся, в частности: локальные и удалённые терминальные входы пользователей; ВСЕ выполняемые пользователями команды; операции ввода-вывода, выполняемые определёнными прикладными программами и пр. В состав системы аудита входят также средства анализа созданных журналов с целью поиска информации о событиях с определёнными признаками. Обзор средств аудита содержится в лекции 16.