Несмотря на широкую распространенность антивирусных программ, вирусы продолжают «плодиться». Чтобы справиться с ними, необходимо создавать более универсальные и качественно новые антивирусные программы, которые будут включать в себя все положительные качества своих предшественников. Защищенность от вирусов зависит и от грамотности пользователя. Применение вкупе всех видов защит позволит достигнуть высокой безопасности компьютера, и соответственно, информации. Помимо защиты всех источников проникновения вредоносных программ, крайне важно периодически проводить проверку компьютера на присутствие вирусов. Это необходимо делать для того, чтобы исключить возможность распространения вредоносных программ, которые не были обнаружены компонентами защиты из-за, например, установленного низкого уровня защиты или по другим причинам.

4.2. Проактивная антивирусная защита – функции и возможности

4.2.1. Поведенческий контроль (Behavior Control)

Проактивная защита. Компоненты поведенческого контроля (Behavior Control) осуществляют мониторинг действий всех приложений в системе и блокируют действия, которые угрожают безопасности системы и ее пользователям. *Также называют: Проактивная защита (Proactive Protection, Proactive Defense), активный вирусный контроль (Active Virus Control), защитный экран от вторжений (Intrusion Guard), основная система предотвращения вторжений (HIPS, Host-based Intrusion Prevention System), поведенческий экран (Behavioral Shield), превентивная защита.*

Поведенческий анализ содержит базу данных наборов правил, которые определяют, какие действия должны быть разрешены или заблокированы для каждой программы. Система защиты выполняет контроль и прекращает работу программ, которые могут выполнить потенциально опасное действие. Если существует правило, которое определяет конкретную ситуацию, оно используется для того, чтобы либо разрешить, либо заблокировать действие.

Если какое-либо действие решено заблокировать, исполнение программного потока модифицируется таким образом, чтобы действие не выполнялось и все параметры приложения меняются для того, чтобы гарантировать безопасность; если действие программы разрешено набором правил, его выполнение происходит без изменений со стороны защиты. Иногда не существует определенного правила для действия программы в базе данных. В таких случаях, в зависимости от настроек компонентов поведенческого контроля, пользователю либо предлагается принять решение, либо действие исполняется или блокируется в автоматическом режиме на основании информации эвристического анализа.

Поведенческий контроль не сканирует файлы приложений перед их выполнением, следовательно, невозможно определить вредоносное ПО до запуска приложения. Тем не менее эта опция антивирусных программ позволяет эффективно

блокировать опасное поведение и, таким образом, предотвращать повреждения и защищать систему от известных и неизвестных вирусов.

Другие компоненты, например антивирусный движок (Anti-virus Engine), могут быть тесно связаны с поведенческим контролем и критически необходимы для его правильной работы. Основное направление развития разработки поведенческого контроля связано с возможностью определения степени опасности конкретного приложения до его запуска, даже если оно является неопознанным.

4.2.2. Режимы работы поведенческого контроля

Аналогично политикам безопасности фаервола (Firewall Policy), которые определяют принятие решений с помощью компонента контроля программ (Program Control), функция поведенческого контроля может иметь несколько режимов работы. Большинство антивирусных программ не предоставляют отдельных настроек для поведенческого контроля, в таких решениях эти настройки едины для фаервола (Firewall) и модуля поведенческого контроля. Некоторые антивирусные решения позволяют конфигурировать эти две функции раздельно, однако доступные режимы работы строятся на тех же принципах, что и политика безопасности фаервола:

- **режим обучения** (Learning mode) – поведенческий контроль в автоматическом режиме создает новые наборы правил для действий приложений;
- **интерактивный режим** (Interactive mode) – при спорных ситуациях появляется оповещение, и пользователю предлагается принять решение;
- **тихий режим** (автоматический режим, Silent mode) – все действия опции происходят в автоматическом режиме.

Некоторые антивирусы позволяют установить степень защиты для поведенческого контроля. Эта настройка определяет, какие действия приложений считаются потенциально опасными. На низших уровнях защиты приложениям позволяется свободно выполнять практически все возможные действия, за исключением самых опасных. На высших уровнях защита является более жесткой, программы находятся под тщательным наблюдением. Иногда даже абсолютно безопасные действия приложений могут блокироваться в таких случаях.

В некоторых случаях есть возможность настройки конкретных реакций на каждое потенциально опасное действие. Разрешить (Allow), запретить (Block) или спросить (Prompt) – основные доступные опции, которые означают, что конкретное действие может быть разрешено, заблокировано или требуется решение пользователя.

Настройки поведенческого контроля по умолчанию подразумевают использование преимущественно автоматических действий и меньшего вмешательства пользователя. В таких случаях режим безопасности часто называется «оптимальный». Это означает, что большинство расширенных функций поведенческого контроля отключены и только основные способы обнаружения вредоносного ПО активны. Если пользователь хочет обезопасить себя от самых сложных видов атак, нужно активировать соответствующую опцию. Ее название может отличаться у различных вендоров, она может называться «расширенный мониторинг событий» (Advanced events monitoring), «антиутечка» (Anti-leak). Обычно для этой

технологии используются уникальные названия под зарегистрированными торговыми марками.

4.2.3. Использование песочницы (Sandbox) как изолированной программной среды

Во время работы автоматических режимов антивирусная программа может разрешить потенциально опасное действие (опция «Разрешить все» (Alow All/Alow Most) активирована) или заблокировать абсолютно безопасные действия программы (опция «Заблокировать все» (Block All/Block Most) включена). Работа автоматических режимов неидеальна, а использование интерактивного режима предполагает наличия у пользователя определенной квалификации для принятия решений. Кроме того, большое количество вопросов в интерактивном режиме может раздражать пользователя. В этом случае альтернативой может послужить использование изолированной программной среды (sandbox).

Пример работы песочницы Sandboxie. Антивирусные продукты, которые содержат в своем инструментарии песочницу, обрабатывают все неизвестные или подозрительные программы специальными методами, которые гарантируют, что они не принесут вреда системе. Создается специальная среда, называемая песочницей, которая выглядит для запускаемых внутри приложений как настоящая система. Программы могут свободно управлять объектами только в песочнице, их действия недоступны для реальной системы (например, изменение записей системного реестра).

Изолированная программная среда гарантирует, что опасные действия не навредят ОС, запускаемое в ней приложение не может определить, где именно оно выполняется. Если привести пример с внесением изменений в реестр, то приложение пытается прочитать значение записи после сделанных изменений, в это время песочница возвращает измененные значения, несмотря на то, что в действительности системный реестр оказывается нетронут. Существует несколько причин, почему нельзя создать идеальную песочницу и почему некоторые критические действия постоянно блокируются в безопасной среде. Степень эффективности изолированной программной среды определяется возможностью ее распознания со стороны вредоносного ПО. Чем песочница менее заметна запускаемому в ней приложению, тем лучше.

Надежные программы всегда запускаются вне изолированной программной среды, что позволяет им выполнять любые требуемые для нормальной работы операции. Когда на компьютер устанавливается новое неизвестное ПО и изолированная программная среда запрещает какие-либо действия приложению, пользователь может добавить это приложение в список исключений. Некоторые антивирусные программы имеют в своем арсенале песочницу как отдельную функцию поведенческого анализа. Эти продукты позволяют, отключив изолированную среду, по-прежнему контролировать действие программ. Другие антивирусные решения встраивают песочницу в компоненты поведенческого контроля. Также существуют пакеты безопасности, которые позволяют настраивать, в каких случаях действия приложений должны быть автоматически заблокированы, а в каких должно приниматься решение на основании текущих настроек политики безопасности.

4.2.4. Потенциально опасные действия и процедуры (Potentially Dangerous Actions and Techniques)

Потенциально опасные действия, различаемые современными антивирусными решениями, могут быть разделены на несколько групп. В [1] рассказано о самых основных действиях, контролируемых антивирусами:

Сессия динамического обмена данными (DDE communication) – DDE является межпроцессорным методом связи, позволяющим одновременно запускать две или несколько программ. Серверное приложение, использующее DDE, может получать данные от клиентского приложения и отвечать ему. Некоторые приложения, например Internet Explorer, позволяют другим приложениям осуществлять контроль, используя команды динамического обмена. Эта особенность может использоваться вредоносным ПО для маскировки опасных действий под достоверные источники.

Контроль доступа объектной модели программных компонентов (COM Access Control), контроль автоматизации протокола OLE (OLE Automation Control) – технология автоматизации OLE заменяет DDE. Это более расширенный механизм межпроцессного взаимодействия, основанный на объектной модели программных компонентов. Множество важных системных служб обеспечивают интерфейсы для приложений с помощью технологий COM/OLE. Когда интерфейс используется вирусом, складывается впечатление, что мы имеем дело с доверенной службой, а не потенциально опасной.

Клиентские службы вызова удаленных процедур и системы динамических доменных имен, запрос прикладного программного интерфейса системы динамических доменных имен (DNS/RPC Client Services, DNS API Request) – некоторые системные службы, такие как клиент DNS, доступны с помощью технологий, называемых «вызов удаленных процедур», «вызов локальных процедур» или «расширенный вызов локальных процедур». Эти процедуры используются для межпроцессного взаимодействия. Так же как и вышеупомянутые технологии, эти службы могут быть атакованы вредоносным ПО. Мониторинг связанных взаимодействий может предотвратить злонамеренное пользование этими службами.

Контроль программных окон, контроль сообщений Windows (Application Window Control, Windows Messages) – оконные сообщения являются другим механизмом межпроцессного взаимодействия, а также одним из наиболее используемых пользовательских графических интерфейсов приложений. Они могут часто подвергаться злонамеренному использованию вредоносным ПО. Используя оконные сообщения, возможно имитировать основные действия пользователя, например клик кнопкой мыши. Пока приложение имеет графический интерфейс, основанный на технологии оконных сообщений, оно может быть атаковано вредоносным ПО посредством этого метода.

Внедрение кода, внедрение процесса в системную память, межпроцессорный доступ к памяти (Code Injection, Process Memory Injection, Interprocess Memory Accesses) – внедрение кода в другой процесс, запущенный в системе является простым методом выполнения вредоносного кода под маской доверенного процесса. Вирус может быть ознакомлен с ограничениями поведенческого контроля и для обхода защиты

может внедрять код в надежный процесс, чтобы иметь возможность произвести вредоносные действия. Защита доверенных процессов от внедрения кода является самой главной в поведенческом анализе современных антивирусных продуктов.

Внедрение библиотек DLL (DLL Injection, Binary Planting) – внедрение библиотеки DLL схоже с внедрением вредоносного кода. Результат успешной атаки идентичен – выполнение вредоносного кода посредством доверенного приложения. Различие в том, что в случае внедрения DLL целый модуль загружается в подвергающийся атаке процесс, в то время как внедрение кода подразумевает, как правило, включение небольшой части кода. Внедрение библиотек является простым приемом для разработчиков вирусов, однако эта методика легко определяется антивирусными программами.

Запуск приложений с поддержкой сетевого обмена данными, запуск процесса, родительское управление процессом (Network-enabled Application Launch, Process Launching, Parent Process Control) – в ОС Windows родительский процесс может контролировать дочерние процессы либо с помощью задания определенных команд, либо используя методы, связанные с внутренней функциональностью процесса. Эта особенность представляет еще один метод атаки доверенного процесса вредоносным ПО. Антивирусные программы осуществляют мониторинг цепочки родительских процессов: либо всех запущенных в системе, либо только доверенных.

Завершение процесса (Process Termination) – завершение процесса и схожие виды атак (завершение потока, попытки критического завершения процесса или потока) предполагают частичное повреждение или полное отключение антивирусной защиты. Цели атаки в данном случае – процессы антивируса. Фактический результат успешной атаки зависит от реализации конкретного антивирусного продукта. Атака может привести к нестабильности, зависаниям, критическим ошибкам или отключению некоторых функций безопасности. Некоторые антивирусы могут распознавать повреждение своих компонентов и блокируют ПК для предотвращения дальнейших вредоносных действий.

Низкоуровневый доступ к сети, прямой доступ к сети (Low-level Network Access, Direct Network Access) – большинство антивирусов способны отлично справляться с контролем основного сетевого трафика, такого как веб-серфинг, сообщения e-mail, но появляются проблемы, когда дело касается протоколов специального назначения. Нередки случаи, когда антивирусы позволяют взаимодействие с вебсайтами (при использовании гипертекстового протокола передачи – HTTP) только доверенным источникам, в то время как передача данных посредством протокола управления сетевыми сообщениями (ICMP) происходит бесконтрольно в автоматическом режиме. Таким образом, вредоносные программы, использующие альтернативные методы передачи данных, менее уязвимы для современных антивирусных решений.

Прямой доступ к диску (Direct Disk Access) – основной способ доступа к данным на жестком диске включает системные функции, которые работают с файлами и директориями. Ранние версии Windows позволяют приложениям напрямую обращаться к диску и данным на нем. Такой метод доступа к данным на диске позволяет обходить основные способы защиты директорий. На ОС Windows Vista и более поздних ОС Windows эта процедура ограничена и менее уязвима для вредоносных атак.

Доступ к оперативной памяти, прямой доступ к памяти (Physical Memory Access, Direct Memory Access) – каждый работающий процесс в системе имеет свою собственную память, недоступную другим приложениям по умолчанию. В случаях, когда требуется удаленный доступ к памяти, система делает это возможным с помощью специальных функций. В то же время антивирусная система осуществляет контроль данного правила доступа. Ядро ОС также имеет собственную память, недоступную другим приложениям. Как бы то ни было, в старых ОС Windows была возможность доступа к объекту, который затрагивает всю память, включая область системного ядра. Это позволяло вредоносному ПО обходить основные механизмы доступа к памяти. В Windows Vista и более поздних системах данная опция запрещена.

Установка драйверов устройств, инициализация драйвера (Device Driver Installation, Driver Load) – Приложения, работающие в ОС Windows, имеют некоторые ограничения, особенно касающиеся использования ресурсов аппаратных средств, таких как оперативная память, жесткий диск, устройства ввода и вывода и т.д. Когда приложение стремится использовать аппаратное средство, оно обращается к системному ядру, которое может либо разрешить, либо запретить конкретное действие. Этот механизм отлично работает с программным кодом, работающим в так называемом пользовательском режиме. Код системного ядра в свою очередь работает в так называемом режиме ядра, который позволяет любой доступ к аппаратным средствам без ограничений. Код системного ядра может обходить все виды защиты, включенные в ОС или предоставляемые сторонними программами. Приложение, работающее в пользовательском режиме, может загрузить драйвер устройства, код которого работает в режиме системного ядра. Вот почему вредоносные драйвера не должны загружаться, и необходим постоянный контроль за этим. На 64-битных системах Windows этот метод практически непригоден для использования вредоносными программами из-за запроса цифровой подписи каждого драйвера, работающего в режиме ядра.

Установка служб (Service Installation) – Системные службы в ОС Windows – специальные программы, которые могут работать, даже когда завершен сеанс пользователя. Они являются более приоритетными по сравнению с обычными приложениями, не требуют прямого взаимодействия с пользователями и могут запускаться автоматически во время загрузки системы. Некоторые службы не имеют своих собственных процессов и размещаются в других схожих службах внутри специальных процессов. Службы являются очень простым способом для вредоносного ПО, чтобы закрепиться в системе. Антивирусные программы также обычно имеют одну или несколько служб. Вредоносные программы могут отключить важнейшие компоненты антивируса, если не контролировать постоянно установку системных служб. Более того, Для установки драйверов, работающих в режиме ядра, используется тот же интерфейс, что и для установки системных служб.

Доступ к файлу HOSTS – файлу базы данных доменных имен (HOSTS File Access). HOSTS файл – специальный файл, содержащий соответствия сетевых имен и IP-адресов. Говоря общими словами, сетевые имена – это домены, а связи между доменом и IP-адресом определяются с помощью протокола системных доменных

имен. Как бы то ни было, именно файл HOSTS используется для перевода сетевых имен, включая домены, в IP-адреса. Таким образом, с помощью файла HOSTS возможно перенаправить домен к произвольному IP-адресу. Основной прием вирусов заключается в перенаправлении серверов обновлений антивирусной программы к несуществующим адресам, что парализует возможность обновления антивируса. Другой прием используется для фишинга – перенаправления домена различных электронных платежных систем к вредоносным серверам, которые выглядят идентично оригинальному сайту, и осуществления кражи конфиденциальных платежных данных.

Активные изменения рабочего стола (Active Desktop Changes) – ранние версии ОС Windows имели возможность внесения активного содержимого пользователем на рабочий стол. Эта опция позволяла создавать полностью настраиваемые рабочие столы. Активный рабочий стол может злонамеренно использоваться вредоносным ПО под маской доверенного приложения проводника Windows. Windows Vista и более поздние системы не имеют поддержку активного рабочего стола.

Папки автозагрузки и автозапуска (Autoruns, Autostart Locations) – Приложение имеет множество способов для установки в ОС с последующим автозапуском при перезагрузке системы. Некоторые из этих способов позволяют заражать различные системные процессы вредоносной библиотекой DLL, т.е. выполнять внедрение DLL. В общем случае вредоносные программы используют несколько папок автозагрузки для того, чтобы обосноваться в системе.

Регистрация вводов с клавиатуры, кейлоггинг (Keylogging, Keyboard Logging) – наблюдение за действиями пользователя является еще одной популярной деятельностью вредоносным программ. Методы регистрации клавишного ввода позволяют получить информацию, которую пользователь вводил в другое приложение с помощью клавиатуры. Использование этих методик позволяет воровать пароли, введенные в браузере, почтовом клиенте или клиенте обмена текстовыми сообщениями. Некоторые приемы кейлоггинга основываются на внедрении DLL или захвате оконного интерфейса.

Захват изображений с экрана и логгинг буфера обмена (Screen and Clipboard Logging) – скринлоггинг и регистрация буфера обмена также используются для кражи точной конфиденциальной информации. Выполнение снимков с экрана может быть использовано для кражи данных кредитной карточки, введенных на безопасной веб-странице в браузере. Логгинг буфера обмена позволяет украсть данные, которые пользователь использует для копирования в ОС Windows. Многие пользователи переносят конфиденциальную информацию, такую как пароли, через буфер обмена. Обычно это случается, когда веб-приложение запрашивает сложные пароли. С одной стороны, использование сложных паролей является необходимостью, т.к. они менее уязвимы для взлома, но с другой стороны пользователь, использующий подобные пароли в разных приложениях, физически не в состоянии их запомнить и использует программу для хранения паролей или просто текстовый файл для копирования и вставки пароля в соответствующую форму.

Захватчик окон, захват системных событий (Window Hooking, Windows and WinEvent Hooks) – захват оконных сообщений Windows и так называемых системных событий позволяет ОС предложить ряд специализированных прикладных

программных функций для программ с целью мониторинга оконных сообщений и сформированных уведомлений о системных событиях. Эти функции также могут быть использованы вредоносным ПО для внедрения зловредных действий, таких как внедрение DLL-библиотек или кейлоггинг.

4.2.5. Управление компонентами (Component control)

Каждое приложение использует один или несколько исполняемых модулей, которые иногда называют компонентами. Основной модуль – как правило, файл с расширением .exe, которые предполагает загрузку некоторых динамически связанных библиотек (файлов с расширением .dll), находящихся в той же директории. Основные приложение используют библиотеки ядра системы: Kernel32.dll, KernelBase.dll, ntdll.dll, Advapi32.dll, user32.dll и другие. Множество программ используют сторонние библиотеки, которые устанавливаются в систему вместе с основным программным модулем. Файлы .dll могут загружаться в память либо во время инициализации приложения, либо во время запроса определенной функциональности в приложении.

Таким образом, каждое приложение имеет определенный набор файлов библиотек, которые загружаются в память и от которых зависит его работа. Контроль компонентов (Component Control) определяет эти зависимости и контролирует загрузку модулей в процесс приложения. Когда вредоносное ПО пытается внедрить свою библиотеку DLL в другой процесс, компонентный контроль распознает и запрещает это опасное действие.

Компонентный контроль также гарантирует неприкосновенность достоверных безопасных модулей. Любые попытки изменить файлы надежных известных модулей могут быть распознаны и заблокированы. Это относится как к главным исполняемым файлам, так и к файлам динамически связанных библиотек.

4.2.6. Защита переносных мультимедийных устройств (Removable Media Protection)

Основная функциональность современных антивирусных программ по части защиты переносных мультимедийных устройств (USB-флешки, внешние HDD-диски) предполагает отключение функции автозагрузки или автозапуска. Когда переносное устройство включается в компьютер, а его коренная директория содержит файл Autorun.inf, сторонняя программа может запуститься системой. Это может привести к незаметному заражению компьютера.

Большинство антивирусных решений также определяют специальный набор правил для всех программ, расположенных на переносных устройствах. Предполагается, что файлы на переносных накопителях могут появиться с других ПК, инфицированных и не оснащенных достаточным уровнем безопасности. Вот почему программы на переносных устройствах считаются по умолчанию потенциально опасными и их действия строго ограничены. Некоторые пакеты безопасности могут распознавать программы с электронной подписью от надежных источников и не ограничивать действия таких приложений.

4.2.7. Самозащита (Self-protection)

Поведенческий анализ также отвечает за одну из самых критических функций – самозащиту антивирусной программы. Любая антивирусная защита может оказаться бесполезной, если вредоносное ПО может отключить ее. Современные антивирусные программы защищают все свои компоненты от вирусной угрозы так, чтобы они не могли быть отключены или повреждены. Самозащита предполагает защиту программных процессов и потоков, файлов и директорий, записей реестра и их значений, установленных системных драйверов и служб, интерфейсов COM и других ресурсов, созданных антивирусом и доступных для других процессов в системе.

Предотвращение инфицирования самых важных процессов является жизненно необходимым для любой антивирусной программы. Множество пакетов безопасности полагаются на постоянные обновления их антивирусной базы. Процесс обновления разрабатывается максимально неуязвимым для вредоносного ПО, чтобы вирус не мог остановить загрузку или установку обновлений или загрузить подменные файлы обновлений.

Самозащита, как правило, включена в основной набор правил поведенческого анализа, которые запрещают управление ресурсами антивирусного продукта. Самозащита может идти отдельно от модуля безопасности, управляющего сторонними программами. Во втором случае компоненты антивируса лучше защищены, чем любое другое приложение в системе. Оба подхода имеют место в современных антивирусных решениях.

4.3. Иммунный подход к защите информационных систем

4.3.1. К проблеме уязвимости операционных систем

Выше мы кратко рассмотрели основные принципы организации и применения антивирусных программ, которые позволяют обнаруживать и уничтожать различные вирусы, восстанавливать поврежденные данные, а также функции проактивной защиты, которая позволяет обнаружить новую вредоносную программу еще до того момента, когда она успеет нанести вред. В завершение этой главы мы рассмотрим еще одно перспективное направление обеспечения кибербезопасности – операционные системы с «кибериммунитетом».

В качестве введения в проблему ниже приведем ряд общеизвестных фактов.

Сегодня в мире существует несколько сотен различных операционных систем. Обычно ОС классифицируют по *базовой технологии* (UNIX – подобные), *типу лицензии* (проприетарные или открытые), *по назначению* (универсальные, ОС для встроенных систем, ОС РДА, ОС реального времени, ОС для серверов или для рабочих станций), *устаревшие* и *современные, исследовательские*, а также по множеству других признаков.

Например, только у Microsoft были разработаны такие основные *устаревшие* версии, как MSX-DOS, MS-DOS, Xerix, Microsoft Windows (Windows 1.0, Windows 2.0, Windows 3.0, Windows 9x, Windows 95, Windows 98, Windows Me, Windows NT,