Информационная безопасность и язык программирования C_s

М. А. Малков (Москва)

Безопасный язык программирования C_s и операционная система, созданная компилятором этого языка, обеспечивают полную защиту от вредоносных программ и кибер-атак. Система, состоящая из C_s и его операционной системы, открыта только для программ, созданных компилятором C_s или удовлетворяющих стандартам C_s. Имеется робот, преобразующий все существующие программы к этим стандартам.

Ключевые слова: информационная безопасность, языки программирования, вредоносные программы, кибер-атаки.

Введение

«Кто владеет информацией тот владеет всем миром» (У. Черчиль, Н. Ротшильд). Эта сентенция отражает роль информации в нашей цивилизации — современная цивилизация не может существовать без информации. На приобретение, распространение и защиту информации тратятся огромные деньги. Информация не может существовать без защиты.

Значительная часть преступлений совершается в информации. Ущерб только от компьютерных преступлений составляет миллиарды долларов в год.

На защиту информации тратятся огромные средства. Тем не менее полная защита информации не существует сейчас и не будет существовать в будущем. Но полная защита от вирусов и кибератак существует. Система, реализующая такую защиту , приведена в данной статье. Основу этой безопасности составляет предлагаемый безопасный язык программирования C_s и операционная система, созданная компилятором C_s. Программы, созданные компилятором C_s, имеют область кодов, защищенную от изменений любыми кодами, и область данных, не содержащую кодов. Операционная система C_s работает только с программами, созданными компилятором C_s или удовлетворяющими стандартам C_s.

Операционная система осуществляет конвертацию к стандартам C_s информации, поступающей из интернета. Система, состоящая из C_s и его операционной системы, обслуживается сервером разработчика системы. Сервер обеспечивает всех пользователей системы интернетом и защитой от кибер-атак типа DDOS.

Внутренняя защита

Под внутренней подразумевается защита программы от ее ошибок и от вмешательства других программ компьютера или программ компьютеров корпоративной сети. Это достигается средствами языка, компилятора и операционной системы.

Язык C_s. Язык C_s наследует синтаксис C++. Но из C++ удалены операторы, нарушающие безопасность компьютера. Удалены указатели, так как они дают неограниченный доступ к любому участку памяти (удаление указателей реализовано уже в языке Ruby [1]). Операторы беспорядоченного выделения и освобождения участков памяти тоже удалены, так как память для данных и программ выделяется автоматически при входе в блок выполняемой программы и освобождается при выходе из блока. Исключением являются объекты, объявленные с спецификаторами памяти *static* и *extern*. Все это позволяет выделять непрерывные но изолированные участки памяти для программ и данных и защитить память программ от их изменения кодами, а память данных от внесения в нее кодов. Коды не вносятся в память данных, так как это бесполезно — данные защищены от использования в качестве программ, и извлеченные из этих данных коды считаются данными, следовательно эти коды не могут исполняться. Операторы *goto*, *while*, $do - while$ удалены, их функцию выполняют операторы for, *continue*, *break*. Это позволяет реализовать защиту от зацикливания программ путем добавления оператору for четвертого параметра, устанавливающего максимальное число циклов. По умолчанию это число равно 1 000 000. При объявлении массива информации можно указать объем памяти для него со знаком «+». Тогда операционная система увеличивает объем этой памяти при превышении отведенного объема, но не более чем в три раза.

Язык C_s допускает гипертексты, в частности гипертексты HTML.

Существуют и другие особенности языка C_s, но они не предназначены для защиты информации.

Компилятор C_s. Компилятор преобразует программы на языке C_s в коды и ищет ошибки в тексте программ. Компилятор осуществляет относительную адресацию кодов и данных. Он также обеспечивает защиту кодов от их изменения кодами и защиту данных от ошибок адресации.

Для этого каждый массив информации добавляется данными о текущем объеме этого массива. При попытке чтения за пределами объема осуществляется прерывание программы. При попытке записи за пределами объема, отведенного при объявлении массива, осуществляется прерывание программы или обращение к операционной системе для увеличения этого объема. Информация о текущем объеме массивов позволяет оперировать массивами целиком, без организации многочисленных циклов перебора элементов массива. Это увеличивает надежность программ.

Ошибки программ, не обнаруженные компилятором, анализируются разработчиком системы. Разработчик вносит изменения в компилятор для устранения этих ошибок. Это позволяет свести к минимуму число необнаруженных ошибок программ.

Компилятор ищет ошибки и угрозы в гипертекстах тоже. Результаты компиляции помещаются в библиотеку пользователя.

Операционная система. Операционная система наследует Windows 10, многие компоненты которой, не обеспечивающие информационную защиту, удалены. В частности отсутствует функция администратора и отсутствуют любые вмешательства в работу программ. Операционная система осуществляет только выполнение программ в мультизадачном режиме, выделение памяти программам и обращение к внешним устройствам. Память выделяется непрерывной областью. При запросе на увеличение памяти операционная система увеличивает ее без сдвига программ, если это возможно, или с сдвигом программ, одновременно осуществляя уборку мусора. Операционная система выделяет программам внешнюю память для данных и блокирует обращение к внешней памяти других программ. Программы читаются и записываются только в библиотеки программ.

Программное обеспечение операционной системы хранится в ее библиотеке. Запись в эту библиотеку доступна только разработчику системы, чтение доступно только операционной системе. Существует общесистемная библиотека. Запись в такую библиотеку доступна только разработчику системы, чтение доступно любому пользователю. Существуют корпаритивные библиотеки. Запись в такую библиотеку доступна только администратору этой библиотеки, чтение доступно пользователям корпоратива. Администратор может выделять подбиблиотеки, доступные для чтения и записи отдельным кругам пользователей. Существуют библиотеки пользователей. Каждый пользователь имеет доступ только к своей библиотеке.

Все библиотеки хранят только программы, созданные компилятором C_s или удовлетворяющие стандартам C_s.

Аналогичная архитектура существует для хранилищ данных.

Компьютерный робот. Реализация внутренней защиты приводит к возникновению новых проблем. Основная проблема обусловлена тем, что система закрыта для программ, не удовлетворяющих стандартам C_s. Данную проблему решает компьютерный робот, предназначенный для конвертации этих програм к стандартам C_s. Дополнительно к конвертации компьютерный робот генерирует тексты этих программ на языке C_s. Это позволит разработчику совершенствовать эти программы с целью обеспечить системный подход к работе с ними. Системный подход позволяет работать с новой программой без изучения ее документации. Тем не менее документация доступна в виде электронного справочника, тоже удовлетворяющего системному подходу. Этот справочник облегчает освоение документации.

Внешняя защита

Внешней является защита от вредоносных программ и кибер-атак в интернете [2, 3]. Внешнюю защиту обеспечивают операционная система и сервер разработчика.

Операционная система. Операционная система обеспечивает защиту данных и программ от внешних угроз точно также, как и защиту от внутренних угроз — данные и программы любого пользователя доступны для чтения и записи только пользователю. Операционная система выделяет внутреннюю и внешнию память для программы, работающей с интернетом. Данная программа создана компилятором C_s, что позволяет ей блокировать выход за пределы этой памяти. Кроме того, эта программа осуществляет некоторые функции операционной системы: выделяет память каждому источнику интернетовской информации, защищает эту память от других источников информации и блокирует передачу данных и программ источником информации не своему сайту. Таким образом, исполняемые коды источника не могут читать информацию из и писать информацию в память других источников и в память пользователя. Пользователь читает и слушает информацию источника, а также просматривает его графику. Вместо копирования данных и программ источника пользователь осуществляет только конвертацию их к стандартам C_s, эта конвертация, в частности, удаляет коды, генерирующие другие коды, и блокирует чтение и запись за пределы выделяемой памяти.

Информация, отсылаемая пользователем любому сайту, не содержит зловредных программ (см. подраздел 2.2). Но объем такой информации ограничивается, если он создает помехи объекту, получающему эту информацию. Данное ограничение не позволяет пользователю осуществ-

лять кибер-атаки типа DOS. Других ограничений операционная система не делает.

Сервер разработчика. Разработчиком является большой коллектив программистов, так как им необходимо создать и сопровождать огромный объем программ. Создание программ осуществляет компьютерный робот, разыскивающий программы в интернете. Он работает круглосуточно из-за огромного объема этих программ. Робот помещает каждую программу в папку, отведенную владельцу этой программы. Папки создаются в системной библиотеке, которая доступна любому пользователю. Все программы безопасны. В дальнейшем разработчик преобразует эти программы для обеспечения системного подхода.

Сервер является интернет-провайдером верхнего уровня для пользователей системы C_s. Нижний уровень образует сеть локальных интернет-провайдеров. Все пользователи системы C_s обслуживаются только одним из локальных провайдеров, другие провайдеры заблокированы. Каждый локальный провайдер имеет межсетевой экран [3, 4], программы которого созданы разработчиком. Кроме обычных функций, межсетевой экран обеспечивает защиту от кибер-атак типа DDOS. Для этого регистрируются все клиенты провайдера и наиболее часто используемых ими сайты, включая почтовые серверы, если эти сайты не принадлежат клиентам. Клиенты исключены, так как их программное обеспечение не содержит вредоносных программ. Следовательно, клиенты не могут быть участниками кибер-атак.

При кибер-атаке на одного из клиентов блокируются все незарегистрированные сайты, посылающие пакеты клиенту. Если и этого недостаточно, то блокируются сайты с наибольшим числом пакетов. После завершения атаки все блокировки отключаются. В результате работа клиента осуществляется в обычном режиме во время атаки.

Список литературы

[1] Thomas D., Fowler C., Hunt A. Programming Ruby: The Pragmatic Programmer's Guide, second edition. — Boston: AddisonWesley, 2004.

[2] Таненбаум Э., Уэзеролл Д. Компьютерные сети, пятое издание. — СПб.: Питер, 2012.

[3] Галатенко В. А. Основы информационной безопасности. Курс лекций. Третье изд. — М: ИНТУИТ, 2006.

[4] Лапонина О. Р. Межсетевое экранирование. — М: Бином, 2014.