Жуковский Павел, 4 курс, 12 группа

«Что такое БИС и с чем ее едят?»

БИС – означает «Безопасность Информационных Систем». Данная дисциплина позволяет понять то, как работают различные информационные системы (например, операционные и файловые системы). Во время изучения данной дисциплины мы можем задуматься о том, насколько уязвимы те или иные компоненты на нашем компьютере, а также то, каким образом мы можем их защитить.

Кстати говоря, система – это совокупность взаимосвязанных и взаимодействующих между собой элементов, которая характеризуется строго определенными функционалом, при этом добавление, удаление, изменение какого-либо элемента и(или) связи между этими элементами приводит к разрушению системы и формированию новой системы (либо к разрушению системы). Это определение я запомнил из самой первой лекции Зубовича Константина Антоновича. Ниже я также буду приводить некоторые определения из лекций Константина Антоновича.

Если мы говорим про **информационную** систему, то эта система представляет из себя систему создания, хранения и передачи данных, другими словами, совокупность взаимосвязанных и взаимодействующих между собой устройств, программных средств генерации, хранения, передачи данных, а также административных установлений, документации и персонала, осуществляющего управление устройствами и программным обеспечением.

Скажем пару слов о программах. **Программа** – это последовательность команд и операторов, которые после расшифровки её вычислительной машиной, может заставить электронно-вычислительную машину выполнять строго определённую последовательность действий. Соответственное, **программное обеспечение** – система взаимосвязанных и взаимодействующих между собой программ.

Программы бывают самые разные. Они бывают как полезными для системы, так и вредными. Например, вирус – это программа (кстати, вредная). Антивирус – это тоже программа (которая направлена на поиск исполнимых модулей с определённой сигнатурой и их удаление). Таким образом, программы внутри системы очень часто взаимодействуют друг с другом. Вместе они образуют систему. Я считаю, что вирусы – это тоже, в каком-то смысле, часть системы, ибо не бывают бочки мёда без ложки дёгтя. Однако для системы можно разработать средства и инструменты для её защиты от этих вирусов.

Если говорить про компьютер, то я бы сказал, что это место, в котором может существовать та или иная информационная система. Если прибегать к строгому определению, то любой компьютер представляет из себя **вычислительную машину** – устройство, которое автоматически и непрерывно производит вычисления, выбирая свои действия из множества заранее установленных команд. Есть также **универсальная вычислительная машина** – вычислительная машина, команды которой записаны в оперативную память и могут быть достаточно быстро заменены в этой операционной памяти другими командами без интерактивного вмешательства человека.

На любом компьютере (вычислительной машине) обязательно присутствует та или иная операционная система. Наиболее популярными операционными системами в наше время являются: Windows (Windows 10, Windows 8, Windows 7, Windows XP и другие), macOS, Linux и другие. Однако существуют и более специфические операционные системы. Например, в военной сфере часто применяются компьютеры со своими операционными системами. Операционная система создаётся таким образом, чтобы она позволяла выполнять те или иные задачи. Популярность Windows, maсOS и Linux обусловлена, наверное, тем, что эти системы хорошо подходят для среднестатистического человека, который просто хочет зайти в Интернет, написать какой-то отчёт, поиграть в какие-нибудь компьютерные игры.

Однако, что из себя представляет операционная система, спросите вы? **Операционная система** – совокупность программных и аппаратных средств, предназначенная для управления устройствами электронно-вычислительной машины и для предоставления пользователю сервиса по управлению этими устройствами. Любая операционная система состоит как минимум из трёх элементов: файловая система, базовая система ввода/вывода и командный язык. Кстати, в рамках операционных систем, «пользователь» представляет из себя совокупность программных средств, которые желают управлять устройствами, электронно-вычислительной машиной (ЭВМ), используя ресурсы операционной системы по управлению этими устройствами.

Стоит отметить, что не существует ни одной такой информационной системы, для которой существовали бы абсолютные методы защиты данных и программ. Любую систему можно взломать, любой системе можно навредить. Вопрос только в том, сколько времени и сил на это потребуется. Вы спросите, почему это так? Любая система на физическом уровне представляет из себя совокупность нулей и единичек, которые ожидают (для дальнейшей правильной работы) соответствующие нули и единички, а это значит, что для любой системы возможно подобрать такую последовательность нулей и единичек (команд), чтобы она продолжала свою работу. Уже с точки зрения физического уровня, любая система уязвима. Всё, что можно сделать, – это усложнить нахождение соответствующих нулей и единичек для злоумышленника. Например, для этих целей можно использовать такие методы защиты, как: шифрование, запутывание, изменение программой самой себя, защита от отладки и другие. Ни один из этих методов (и даже их комбинация), как писалось выше, не сможет гарантировать абсолютную защиту, но, по крайней мере, увеличит количество времени и усилий, которое злоумышленнику придётся затратить, чтобы взломать систему (или тем или иным образом навредить ей).

Однако защищать систему тем или иным методом защиты (будь то шифрование, запутывание или изменение программой самой себя) довольно сложно и далеко не всегда нужно. Обычно, людям лень реализовать какие-либо методы защиты для своих информационных систем. Реализация тех или иных методов защиты системы, как и их взлом, требеут некоторых усилий и времени. Например, в браузерах незащищенные сайты используют протокол http, в то время как защищённые сайты используют протокол https, который подразумевает под собой ещё и некоторое шифрование данных. Конечно, первый протокол проще для реализации, однако второй протокол, хоть и сложнее по времени и усилиям, позволяет шифровать данные, тем самым обеспечивая некоторый уровень безопасности.

Из вышесказанного можно сделать вывод о том, что построение системы для защиты данных оправдано тогда и только тогда, когда стоимость разработки и внедрение данной системы меньше стоимости вреда, который может быть нанесён заказчику системы и за её отсутствие. Например, если мы создали информационную систему, которая нам не очень важна, то мы можем и не защищать её, если суммарное количество усилий и времени на разработку методов защиты этой системы больше, чем максимальный урон, который этой системе могут нанести какие-нибудь злоумышленники. Если приводить примеры из реальной жизни, то, например, в какой-нибудь простой компьютерной сетевой игре пользователи используют пароли для входа в аккаунт. Конечно, этот метод защиты не может обеспечить абсолютную безопасность, особенно если пароль маленький и простой. Это не потому, что разработчики этой простой игры не могут сделать методы защиты сильнее, а потому, что это просто не нужно. Например, в банковских системах, которые работают с деньгами, защита крайне важна. В подобных системах зачастую приходится затрачивать больше усилий на разработку методов защиты, чем на основной функционал. И это оправдано, потому что максимальный урон от взлома подобной системы достаточно велик.

В лекциях Зубовича также приводились следующие два постулата: 1) Тем, что нельзя измерить, невозможно, управлять; 2) Чем ближе к аппаратному уровень реализации системы защиты данных, тем более надёжна сама система защиты данных при прочих равных условиях. Первый постулат говорит нам, что если мы хотим управлять нашей системой, понимать, как она работает в деталях, тем или иным образом влиять на неё и взаимодействовать с ней так, чтобы она работала тем образом, каким мы от неё хотим, то нам необходимо уметь измерять эту систему, нам необходимо понимать, как она работает и как она устроена. Второй же постулат достаточно прост. Уровень реализации той или иной системы можно быть устроен либо ближе к аппаратному, либо ближе к программному. Аппаратный уровень подразумевает собой некое физическое взаимодействие с пользователем этой системе (нажатие на кнопку, поворачивание какого-нибудь рычага, стук по какой-нибудь клавише). Эти действия не могут быть «взломаны» злоумышленником. Однако в программном уровне реализации пользователь взаимодействует с системой при помощи тех или иных команд, однако в этом случае, его действия уже могут быть уязвимы для взлома или, например, шпионажа, так как не существует абсолютно защищённой системы.

За курс лекций по БИС я также узнал, что любую программу можно получить, если заготовить для неё так называемый исходный текст. Исходный текст представляет из себя совокупность некоторых символов, из которых позже может быть получен так называемой объектный модуль. Процесс получения объектного модуля из исходного текста называют **трансляцией**. Кстати говоря, существует много различных методов этой трансляции. Эти методы трансляции часто называют «языками программирования». Среди таких можно выделить, например, Assembler. Суть трансляции заключается в том, что программа трансляции (например, TASM.EXE) переводит инструкции команд, понятные для пользователя, в инструкции команды, понятные для электронно-вычислительной машины. Как только объектный модуль, содержащий в себе совокупность машинных инструкций программы, готов, из него можно получить так называемый исполнимый модуль. Процесс получения исполнимого модуля из объектного модуля называют **линковкой**. Этот процесс можно осуществить, например, с помощью TLINK.EXE. Команда TLINK позволяет получить исполнимый модуль формата \*.exe. В случае, если после команды TLINK будет указан параметр /t, команда TLINK создаст исполнимый модуль формата \*.com. Как только исполнимый модуль получен, его можно загрузить в оперативную память компьютера (вернее, совокупность машинных инструкций, хранящихся в этом исполнимом модуле), после чего программа начнёт своё выполнение.

Учитывая вышесказанное, мы можем использовать язык Assembler, чтобы создавать исходные текста необходимых нам для защиты нашей информационной системы программ (например, антивирусов), а затем, с помощью TASM.EXE, получать из этих исходных текстов соответствующие объектные модули, после чего, с помощью TLINK.EXE, получать из этих объектных модулей соответствующие исполнимые модули. Это позволит нам получить исполнимый модуль программы, которую мы можем в любой момент загрузить в оперативную память нашей информационной системы, обеспечивая тем самым ту или иную защиту.

Я считаю, что предмет БИС – достаточно интересный. Он позволяет понять, что из себя представляет информационная система на достаточно глубоком уровне, а также то, как мы можем навредить этой системе или, наоборот, как мы можем защитить эту систему.