## Понятие защиты программного обеспечения компьютерных систем.

Программные продукты и компьютерные базы данных являются предметом интеллектуаль­ного труда специалистов высокой квалификации. Процесс проектирования и реализации программных продуктов характеризуется значительными материальными и трудовыми за­тратами, основан на использовании наукоемких технологий и инструментария, требует при­менения и соответствующего уровня дорогостоящей вычислительной техники. Это обусловливает необходимость принятия мер по защите интересов разработчика программ и создателей компьютерных баз данных от несанкционированного их использования.

Программное обеспечение является объектом защиты также и в связи со сложностью и трудоемкостью восстановления его работоспособности, значимостью программного обес­печения для работы информационной системы.

Защита программного обеспечения преследует цели:

* ограничение несанкционированного доступа к программам или**их** преднамеренное разрушение и хищение;
* исключение несанкционированного копирования (тиражирования) программ.
* программный продукт и базы данных должны быть защищены по нескольким направ­лениям от воздействия:

1.     человека — хищение машинных носителей и документации программного обеспе­чения; нарушение работоспособности программного продукта и др.;

2.     аппаратуры — подключение к компьютеру аппаратных средств для считывания программ и данных или их физического разрушения;

3.     специализированных программ — приведение программного продукта или базы данных в неработоспособное состояние (например, вирусное заражение), несанкциониро­ванное копирование программ и базы данных и т.д.

Безопасность программного обеспечения (ПО) в широком смысле является свойством данного ПО функционировать без проявления различных негативных последствий для конкретной компьютерной системы. Под уровнем безопасности ПО понимается вероятность того, что при заданных условиях в процессе его эксплуатации будет получен функционально пригодный результат. Причины, приводящие к функционально непригодному результату могут быть разными: сбои компьютерных систем, ошибки программистов и операторов, дефекты в ПО. При этом дефекты принято рассматривать двух типов: преднамеренные и непреднамеренные. Первые являются, как правило, результатом злоумышленных действий, вторые - ошибочных действий человека.

При исследовании проблем защиты ПО от преднамеренных дефектов неизбежна постановка следующих вопросов:

* кто потенциально может осуществить практическое внедрение программных дефектов деструктивного воздействия в исполняемый программный код;
* каковы возможные мотивы действий субъекта, осуществляющего разработку таких дефектов;
* как можно идентифицировать наличие программного дефекта;
* как можно отличить преднамеренный программный дефект от программной ошибки;
* каковы наиболее вероятные последствия активизации деструктивных программных средств при эксплуатации КС.

При ответе на первый вопрос следует отметить, что это: непосредственные разработчики алгоритмов и программ для компьютерных систем. Они хорошо знакомы с технологией разработки программных средств, имеют опыт разработки алгоритмов и программ для конкретных прикладных систем, знают тонкости существующей технологии отработки и испытаний программных компонентов и представляют особенности эксплуатации и целевого применения разрабатываемой КС. Кроме того, при эксплуатации программных комплексов возможен следующий примерный алгоритм внесения программного дефекта: дизассемблирование исполняемого программного кода, получение исходного текста, привнесение в него деструктивной программы, повторная компиляция, корректировка идентификационных признаков программы (в связи с необходимостью получения программы "схожей" с оригиналом). Таким образом, манипуляции подобного рода могут сделать и посторонние высококлассные программисты, имеющие опыт разработки и отладки программ на ассемблерном уровне.

В качестве предположений при ответе на второй вопрос следует отметить, что алгоритмические и программные закладки могут быть реализованы в составе программного компонента вследствие следующих факторов:

* в результате инициативных злоумышленных действий непосредственных разработчиков алгоритмов и программ;
* в результате штатной деятельности специальных служб и организаций, а также отдельных злоумышленников;
* в результате применения инструментальных средств проектирования ПО, несущих вредоносное свойство автоматической генерации деструктивных программных средств.

Для описания мотивов злоумышленных действий при разработке программных компонентов необходим психологический "портрет" злоумышленника, что требует проведения специальных исследований психологов и криминологов в области психологии программирования (психологии криминального программирования, см. раздел 4.4). Однако некоторые мотивы очевидны уже сейчас и могут диктоваться следующим:

* неустойчивым психологическим состоянием алгоритмистов и программистов, обусловленным сложностью взаимоотношений в коллективе, перспективой потерять работу, резким снижением уровня благосостояния, отсутствием уверенности в завтрашнем дне и т.п., в результате чего может возникнуть, а впоследствии быть реализована, мысль отмщения;
* неудовлетворенностью личных амбиций непосредственного разработчика алгоритма или программы, считающего себя непризнанным талантом, в результате чего может появиться стремление доказать и показать кому-либо (в том числе и самому себе) таким способом свои высокие интеллектуальные возможности;
* перспективой выезда за границу на постоянное место жительства (перспективной перехода в другую организацию, например, конкурирующую) с надеждой получить вознаграждение за сведения о программной закладке и механизме ее активизации, а также возможностью таким способом заблокировать применение определенного класса программных средств по избранному месту жительства;
* потенциальной возможностью получить вознаграждение за устранение возникшего при испытаниях или эксплуатации системы "программного отказа" и т.п.

Кроме того, необходимо иметь в виду, что в конструировании вредоносной программы, так или иначе, присутствует притягательное творческое начало, которое само по себе может стать целью. При этом сам "творец" может слабо представлять все возможные результаты и последствия применения своей "конструкции", либо вообще не задумываться о них.

Таким образом, правомерно утверждать, что вредоносные программы, в отличие от широко применяемых электронных закладок, являются более изощренными объектами, обладающими большей скрытностью и эффективностью применения.

Ответы на три последних вопроса можно найти в рамках быстро развивающейся методологии обеспечения безопасности программных средств и оценки уровня их защищенности (разделы 2 и 3).