**Израилов К.Е.**
**Россия, Санкт-Петербург, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А.Бонч-Бруевича**

## ПОИСК УЯЗВИМОСТЕЙ В РАЗЛИЧНЫХ ПРЕДСТАВЛЕНИЯХ ПРОГРАММНОГО КОДА

Несмотря на неослабевающую актуальность поиска уязвимостей в программном коде устройств для всех области, одной из наиболее приоритетных считается телекоммуникационная. Во-первых, это следует из того, что основным предназначением телекоммуникационных устройств является обработка информации, которая зачастую является конфиденциальной. А, во-вторых (что особенно актуально для России), такие устройства поставляются зарубежными производителями без исходного кода – только в виде машинного, что значительно затрудняет анализ кода и последующий поиск.

Перед разработкой методов поиска уязвимостей необходимо четкое понимание причин их появления – как случайных, так и злонамеренных. А с учетом того, что программный код от самого зарождения до окончательной загрузки в рабочее устройство пребывает в различных представлениях, то необходимо учитывать возможности появления уязвимостей в каждом из них. Такое понимание, в частности, позволит выбирать и использовать максимально подходящие способы поиска уязвимостей для каждого представления.

Типичные для телекоммуникационного устройства поэтапные преобразования программного кода в процессе создания конечного продукта проходят через следующие представления; при этом форма представления может быть различной – текстовой, бинарной, в виде блок-схем. Вначале возникает мысленный прообраз кода, отражающий основную идею и описывающий требуемый функционал, характеристики, свойства. Уязвимости в таком прообразе отсутствуют, поскольку на данном этапе невозможно их отличить от особенностей самой идеи.

Используя прообраз, разрабатывается концептуальная модель, вводящая основные понятия, их структуру, взаимосвязь и т.п. В этот момент уже возможно возникновение уязвимостей в виде принципиальных различий созданной модели и первоначальной идеи.

По концептуальной модели разрабатывается архитектура кода, приближенная к будущей реализации; в частности, архитектура может содержать конкретные технологии, парадигмы языков программирования, форматы данных. Ошибки в проектировании архитектуры также приведут к наличию уязвимостей в программном коде. Используя архитектуру, создаются алгоритмы работы ее функционала.

Данный процесс носит творческий характер, что как неизменно связано с появлением в схеме алгоритма случайных ошибок, так и успешно используется для внедрения злонамеренной логики, например программных закладок. Созданные алгоритмы, как правило, переписываются в виде операций на заданном архитектурой языке программирования, создавая так называемый исходный код программы. При достаточной степени детализации алгоритмов и аккуратности выполнения программистом кодирования случайных ошибок можно избежать; тем не менее, на данном этапе возможно (и успешно применяется) внесение в код злонамеренных уязвимостей, хотя и имеющих менее абстрактный уровень, чем в предыдущем представлении.

Далее исходный код преобразуется в ассемблерный с помощью специальных утилит – компиляторов. Возникновение в результате ошибок в инструкциях процессора носит лишь теоретический характер, поскольку это возможно вследствие неверной работы таких утилит, что маловероятно. Машинный код программы получается из ассемблерного в общем виде тривиальным преобразованием инструкций процессора из записанных в текстовом виде в соответствующий им бинарный. Возникновение уязвимостей в этом представлении, аналогично ассемблерному, практически невозможно. Для загрузки на конечное телекоммуникационное устройство выполнение, машинный код, как правило, собирается в единый файл образа, также без каких-либо новых уязвимостей.

Согласно приведенным представлениям можно сделать следующие выводы. Во-первых, преобразования понижают абстрактность используемых в представлениях элементов – от общей идеи через пошаговые алгоритмы до инструкций процессора. Во-вторых, уязвимости могут появляться в различных представлениях, притом имея соответствующий структурный уровень (тип) согласно их абстрактности в программном коде – уязвимости в концептуальной модели и архитектуре (высокоуровневый), алгоритмах (среднеуровневый) и их реализации через операции (низкоуровневый). И, в-третьих, возникшая уязвимость в одном представлении существует и во всех последующих, поскольку происходит лишь ее «размывание» по программному коду. В

се это позволяет утверждать, что поиск уязвимостей каждого типа целесообразно производить по представлению, строящемуся на элементах соответствующего уровня абстракции. В случае же наличия для поиска только машинного кода, обозначенные для поиска представления необходимо из него получить. Также возможным решением может быть получение объединенного представления для поиска всех трех типов уязвимостей.