*D.V. Gurov, V. V. Gurov, M. A. Ivanov, L. I. Shustova*

**The Peculiarities of Teaching Secure Programming in High er School**

*Key words: safe programming, software vulnerabilities, computer learning programs*

The ways of vulnerabilities occurrence in programs are considered. The classification of low level vulnerabilities is included. The methods of protection from such vulnerabilities are analyzed. The education methods for secure programming and the structure of electronic lesson to learn the basics of secure programming are presented.

*Д. В. Гуров, В. В. Гуров, М. А. Иванов, Л. И. Шустова*

ОСОБЕННОСТИ ОБУЧЕНИЯ БЕЗОПАСНОМУ ПРОГРАММИРОВАНИЮ В ВУЗЕ

В настоящее время проблема обеспечения безопасности наряду с обеспечением надежности выходит на первый план при проектировании программной системы. Уязвимости могут появиться в программе двумя основными путями. Это либо специальные закладки, сделанные программистом на стадии разработки программы с целью последующего их злонамеренного использования, либо уязвимости, возникшие из-за недостаточной квалификации разработчика программы, что позволяет в последующем более квалифицированному программисту-злоумышленнику обнаружить и использовать их. В докладе рассматривается направление повышения безопасности программ, связанное со вторым из указанных выше путей появления уязвимостей, — подготовка в вузе специалистов, способных не только написать правильно работающую программу на том или ином языке программирования, но и обеспечить отсутствие (или хотя бы минимизацию) в ней уязвимостей.

В [1] представлена разработанная авторами классификация уязвимостей, названных уязвимостями низкого уровня. Эти уязвимости связаны с особенностями выполнения процесса на уровне взаимодействия системных ресурсов и операционной системы. Для каждого типа уязвимостей даются рекомендации по предотвращению их появления в программе.

Для борьбы с появлением уязвимостей существуют безопасные функции, специальные надстройки компилятора, ограничения на исполнение кода в стеке. Однако количество уязвимостей в рядовых, а зачастую и ответственных программах остается достаточно высоким. Это, на наш взгляд, во многом обусловлено недостаточностью прикладных знаний в этой области у рядовых программистов, занимающихся разработкой конкретных приложений. Если внимательно проанализировать учебные программы, по которым студенты обучаются в вузах, то можно заметить, что специального курса, направленного на изучение основ безопасного программирования, показывающего механизмы появления уязвимостей в ПО и приемы, позволяющие избежать уязвимостей в конечных продуктах, в них нет [2]. В связи с этим на кафедре «Компьютерные системы и технологии» НИЯу МИФИ было определено в качестве одного из приоритетных направлений создание методики обучения студентов основам безопасного программирования.

С целью закрепления материала, получаемого студентами по этому вопросу на лекциях, предлагается включить в состав лабораторного практикума специальный электронный урок, направленный на практическое овладение навыками безопасного программирования. урок содержит теоретический материал, в котором рассказывается о существующих уязвимостях, причинах их возникновения и основных методах защиты от них, подробно описывается метод обнаружения уязвимостей в скомпилированной программе, когда исходные коды недоступны, — вставка ошибки (fault injection).

В качестве исходных данных студенту предлагается интерфейс некоторого консольного приложения, где он может вводить определенные входные значения. Первым заданием для студента является исследование программы на предмет наличия в ней уязвимости по входным параметрам с помощью метода fault injection. Программы, предлагаемые студенту для исследования, генерируются автоматически, но на основании некоторых заданных шаблонов. Это позволяет гарантировать, с одной стороны, наличие и известные свойства уязвимостей в этой программе, а с другой — случайность задания с тем или иным типом уязвимости. Вводя значения, близкие к предельным для тех или иных типов данных, а также значения, содержащие символы-спецификаторы или просто длинные строки, студент должен обнаружить те или иные уязвимости в коде исследуемой программы. Критерием правильности является полнота и точность перечисления полей, связанных с уязвимостями, и указание возможных типов уязвимостей.

На следующем этапе в исходном коде исследуемой программы студент должен отметить строки, которые не соответствуют требованиям безопасного программирования, а затем устранить уязвимость. В данной реализации электронного урока это выполняется путем использования безопасных функций вместо существующих опасных, обязательного использования спецификаторов, использования функций копирования с обязательным заданием количества копируемых байтов, согласованием типов операндов при выполнении операций.

Результаты исправлений в программе проверяются двумя способами — встроенным в электронный урок тестом, в котором хранятся шаблоны безопасных функций, а также внешней программой тестирования ITS4 [3], запуск которой производится из электронного урока. Такая проверка позволяет студенту ознакомиться с принципами работы статических сканеров исходных кодов, предназначенных для обнаружения уязвимостей в программах.

**СПИСОК ЛИТЕРАТУРЫ:**

*1.Гуров В. В., Гуров Д. В., Иванов М. А., Шустова Л. И.* Технология безопасного программирования и особенности ее преподавания в вузе // Дистанционное и виртуальное обучение. 2010. № 9. С. 35—49.

2.Государственные требования к минимуму содержания и уровню подготовки выпускников. [Электронный ресурс]. URL: http://wmdow.edu.rn/wmdow_catalog/pdf2txt?p_id=25226&p_page=2.

3.ITS4: Software Security Tool. [Электронный ресурс]. URL: http://www.cigital.com/its4/.

**REFERENCES:**

*1.Gurov V. V., Gurov D. V., Ivanov M. A., Shustova L. I.* Technology of safe programming and the peculiarities of its teaching in higher school // Distance and virtual learning. 2010. № 9. P. 35—49.

2.State requirements to a minimum of the contents and level of training of graduates. URL: http://window.edu.ru/window_catalog/ pdf2txt?p_id=25226&p_page=2.

3.ITS4: Software Security Tool. URL: http://www.cigital.com/its4/.