Виктор С. Горбатов, Алексей А. Мещеряков
Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, г. Москва, 115409, Россия,
e-mail: VSGorbatov@mephi.ru, ORCID iD 0000-0001-9998-9733
e-mail: aameshcheryakov@mephi.ru, ORCID iD 0000-0001-5148-7734

КУРС ТРЕНИНГА ПО БЕЗОПАСНОЙ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

DOI: http://dx.doi.org/10.26583/bit.2017.2.04

Аннотация. Безопасность является одним из важнейших критериев качества разрабатываемого программного обеспечения. Для получения достаточного уровня защищенности разрабатываемых приложений компании внедряют процессы обеспечения безопасности в жизненных цикл разработки программного обеспечения. Так компании-разработчики сталкиваются с дефицитом кадров, способных решать задачи как по проектированию и реализации ПО, так и по обеспечению их защиты. Данная статья предлагает описание курса тренинга по безопасной разработке программного обеспечения. Курс по безопасной разработке приложений предназначен для совместного обучения студентов, получающих образование как в области проектирования и реализации ПО, так и в области информационной безопасности.

Ключевые слова: программное обеспечение, безопасность приложений, обучение, образование, информационная безопасность.

Для цитирования. **ГОРБАТОВ Виктор. С.; МЕЩЕРЯКОВ, Алексей А.** Курс тренинга по безопасной разработке программного обеспечения. Безопасность информационных технологий, [S.l.], v. 24, n. 2, p. 35-41, june 2017. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/103>. Дата доступа: 23 june 2017. doi:http://dx.doi.org/10.26583/bit.2017.2.04.

Viktor S. Gorbatov, Alexey A. Meshcheriakov
National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashirskoe shosse, 31, Moscow, 115409, Russia,
e-mail: VSGorbatov@mephi.ru, ResearcherID B-5106-2016
e-mail: aameshcheryakov@mephi.ru, ORCID iD 0000-0001-5148-7734

Secure software development training course

DOI: http://dx.doi.org/10.26583/bit.2017.2.04

Abstract. Information security is one of the most important criteria for the quality of developed software. To obtain a sufficient level of application security companies implement security process into software development life cycle. At this stage software companies encounter with deficit employees who able to solve problems of software design, implementation and application security. This article provides a description of the secure software development training course. Training course of application security is designed for co-education students of different IT-specializations.

Keywords: information security, education, secure software development, application security, training.

For citation. **GORBATOV, Viktor. S.; MESHCHERIAKOV, Alexey A.** Secure software development training course. IT Security (Russia), [S.l.], v. 24, n. 2, p. 35-41, june 2017. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/103>. Date accessed: 23 june 2017. doi:http://dx.doi.org/10.26583/bit.2017.2.04.

Введение

Безопасность остается одним из главных приоритетов при разработке программного обеспечения (ПО) на протяжении последних лет. Однако, как показывает практика

большинство разрабатываемых приложений уязвимо к тем или иным атакам. Так исследование безопасности финансовых приложений [1], проведенное компанией Positive Technologies, показало, что все анализируемые системы дистанционного банковского обслуживания содержали уязвимости по крайней мере среднего уровня риска, при этом почти в каждой из систем (90%) были обнаружены критически опасные уязвимости. Схожую статистику по наличию уязвимостей имеют и приложения, написанные для других отраслей экономики и сфер деятельности.

Для получения приемлемого уровня защищенности разрабатываемых приложений компании внедряют процессы обеспечения безопасности в жизненных цикл разработки ПО (SDLC). Внедрение процессов безопасности приложений является сложным и длительным процессом, требующим применения специфических практик безопасности на всех этапах жизненного цикла ПО.

Проблема безопасности приложений находится на стыке ответственности программистов и специалистов по безопасности. Таким образом, от команды по разработке ПО требуется как глубокое понимание угроз и уязвимостей приложений, так и навыки реализации механизмов безопасности, анализа и тестирования кода.

Сегодня система высшего образования не готовит таких специалистов, которые бы хорошо разбирались и в области разработки ПО, и в области его безопасности. Это создает дефицит профессиональных кадров в данной области. Зачастую, компаниям-разработчикам ПО приходится самим дополнительно обучать сотрудников-программистов вопросам безопасной разработки. Сложившаяся ситуация приводит к дополнительным издержкам на разработку, а также создает сложности для донесения до всех программистов информации об актуальных угрозах и важности обеспечения безопасности.

В данной работе описывается процесс создания курса по безопасной разработке приложений для совместного обучения студентов, получающих образование как в области проектирования и реализации ПО, так и в области информационной безопасности. Совместное обучение студентов различных направлений позволяет добиться синергетического эффекта, в результате которого обучающиеся получат новые знания о смежной профессиональной области и навыки взаимодействия между командами разработки и безопасности.

Обзор литературы

Академические работы написанные в области обучения безопасности приложений, направлены преимущественно на построение процесса повышения осведомленности сотрудников в вопросах информационной безопасности и применимы для компаний-разработчиков ПО. Но многие из таких практик могут быть адаптированы для применения в учебном процессе ВУЗа.

Сообщество OWASP (Open Web Application Security Project) подготовило обучающую программу для web-разработчиков [2]. Это четырехчасовой курс, раскрывающий следующие темы: актуальные проблемы безопасности, уязвимости веб-приложений, встраивание в SDLC, обзор хороших практик безопасной разработки, тестирование безопасности.

Правительства многих стран пытается решить проблему обучения информационной безопасности. National Initiative for Cybersecurity Education (NICE) – это государственная образовательная программа кибер-безопасности США [3]. Главной целью NICE является

построение постоянное повышение уровня осведомленности в вопросах безопасности на всех уровнях образования.

В работе [4] показано, что недостаточное изучение вопросов информационной безопасности в учебных программах IT-специальностей может стать основной проблемой, приводящей к возникновению уязвимостей, и предложен подход, предполагающий проведение тренингов для повышения уровня осведомленности.

Стандарты разработки безопасных приложений

The National Institute of Standards and Technology (NIST) оценивает возрастание стоимости исправления дефекта на стадии эксплуатации ПО в 470-880 раз по сравнению с исправлением на стадии сбора требований [5]. Поэтому целесообразнее внедрять процессы обеспечения безопасности во все этапы жизненного цикла, чтобы увеличить вероятность раннего выявления дефектов безопасности.

В области разработки ПО уже подготовлен ряд корпоративных, отраслевых и международных стандартов, описывающих управление процессом создания безопасного ПО:

- ГОСТ Р ИСО/МЭК 27034 [6];
- ISO/IEC TR 24772 [7];
- Microsoft Security Development Life Cycle [8];
- Cisco Security Development Life Cycle [9];
- OpenSAMM [10];
- OWASP CLASP [11];
- BSIMM [12].

Сравнительный анализ представленных выше стандартов проведен в работе [13]. Ввиду многообразия стандартов разного уровня индустрия разработки ПО еще не выработала единого подхода, оформленного в виде единого документа. Поэтому при разработке данного курса мы выделили ряд наиболее применяемых мер и механизмов, рассмотрение которых позволит создать у слушателей целостную картину в изучаемой области:

- Обучение сотрудников безопасной разработке ПО;
- Моделирование угроз безопасности информации, источником которых является ПО;
- Определение требований по разработке безопасного ПО;
- Использование стандарта на оформление исходного кода программы (правил и рекомендаций, направленных на минимизацию количества потенциально уязвимых конструкций в исходном коде);
- Проведение статического анализа исходного кода;
- Проведение динамического анализа ПО;
- Проведение ручной экспертизы исходного кода;
- Проведение анализа уязвимостей;
- Обеспечение безопасной поставки ПО пользователю;
- Устранение уязвимостей ПО, выявленных в процессе функционирования ПО.

Учебная программа

Учебная программа курса тренинга по безопасной разработке приложений включает в себя три модуля:

1. Введение в разработку безопасного программного обучения
   a. Обзор основных угроз ПО.
   b. Безопасность web-приложений.
   c. Безопасность мобильных приложений.
   d. Безопасность десктопных приложений.
   e. Текущие тренды в области безопасности приложений.
2. Разработка безопасного ПО
   a. Формирование требований к безопасному ПО.
   b. Управление конфигурации при разработке безопасного ПО.
   c. Ручная экспертиза кода.
   d. Статический и динамический анализ кода. Фаззинг-тестирование.
   e. Тестирование на проникновение при разработке ПО.
3. Лучшие практики и приемы разработки безопасного ПО
   a. Обзор общих практик и приемов разработки безопасного ПО.
   b. Практики разработки безопасных web-приложений.
   c. Практики разработки безопасных мобильных приложений.
   d. Практики разработки безопасных десктопных приложений.

Задачей первого модуля является знакомство слушателей с актуальным состоянием проблем в рассматриваемой области. Слушатели изучают угрозы и характерные уязвимости безопасности для различных типов приложений. По итогам прохождения данного модуля у слушателя должно сложиться комплексное представление об актуальных угрозах и атаках, которым подвержены современное ПО.

Второй модуль рассматривает меры и механизмы безопасной разработки ПО. В данном модуле слушатели знакомится с моделью безопасности жизненного цикла ПО и получают теоретическую информацию об использовании различных инструментов для обеспечения безопасности приложения.

В третьем модуле изучаются лучшие практики безопасной разработки различных типов приложений. Данный модуль обобщает знания, полученные в первых разделах и дополняет их специфической информацией для каждой конкретной области.

Таким образом, учебная программа курса позволяет сформировать комплексное теоретическое представление об актуальных угрозах, уязвимостях, атаках на ПО, а также о мерах, механизмах и практиках разработки безопасных приложений.

Описание заданий

В ходе прохождения курса по разработке безопасных приложений слушатели выполняют различные домашние задания, связанные с анализом проблем защиты информации, возникающих на каждом из этапов жизненного цикла ПО. Всего курс предусматривает три задания, соответствующие теоретическим модулям:

1. Анализ и формирование требований безопасности к ПО.
2. Выявление уязвимостей в ПО.
3. Реализация модулей безопасности для ПО.

Каждое из заданий ставит своей задачей дать слушателю возможность приобрести практические навыки в области анализа безопасности приложений.

Рис. 1. Структура курса разработки безопасного ПО
Fig. 1. Course structure the secure development

В качестве курсового проекта студентам предлагается разработка какого-либо ПО, например, безопасного текстового чата. Для выполнения такого проекта по мере прохождения курса все студенты делятся на группы, в которые будут входить студенты всех IT-специальностей. Такие группы будут обладать необходимым набором компетенций для выполнения проекта по безопасной разработке. Целью разработки такого проекта является получение практических знаний и навыков в области безопасной разработки. Команды в ходе работы над проектом проходят все этапы жизненного цикла разработки ПО, используя на каждом из них необходимые инструменты обеспечения безопасности приложений. Таким образом, студенты могут оценить на практике эффективность тех или иных инструментов и методов для конкретного проекта. В конце курса каждая команда презентует свой проект перед всеми слушателями. После этого студентов предлагается проведения тестирования на проникновение приложений, разработанных другими командами.

Описание лаборатории

Промышленный процесс разработки ПО предполагает использование различных инструментов: системы управления проектом, системы контроля версий, инструменты тестирования и т.д. Для создания лаборатории были использованы такие инструменты как BurpSuite, OWASP ZAP, Redmine, Fortify [14-17]. Структурная схема развернутой лаборатории курса представлена на рисунке 2.

Рис. 2. Структурная схема лаборатории курса безопасной разработки ПО
Fig. 2. Structural scheme of the laboratory course secure software development

На базе рассмотренного выше стенда развернуты тестовые приложения с преднамеренно внесенными дефектами безопасности. На текущий момент разрабатываются следующие тестовые приложения:

- веб-приложение;
- мобильный клиент для Android.

Разрабатываемые приложения логически связаны между собой в систему имитирующую дистанционную банковскую систему (ДБО). Внесенные дефекты безопасности являются типичными для данных классов ПО и основаны на уязвимостях, найденных в реальных системах. Разрабатываемые приложения анализируются слушателями в ходе выполнения домашних заданий с целью выявления и исправления дефектов безопасности.

Заключение

В данной работе мы рассмотрели курс тренинга по безопасной разработке приложений. В данном курсе было соединено рассмотрение вопросов проектирования, реализации ПО и вопросов его безопасности. Лекции и практические задания раскрывают проблемы безопасности, характерные для различных типов приложений, и рассматривают лучшие практики безопасного программирования. Курсовой проект призван дать слушателям возможность на практике погрузиться в процесс разработки ПО, и оценить необходимость внедрения различных мер и механизмов безопасности на всех этапах жизненного цикла ПО.

Внедрение разработанного курса в учебную программу студентов высших образовательных учреждений всех IT специальностей позволит подготавливать высококвалифицированных специалистов, способных внедрять практики безопасного программирования в организациях.

СПИСОК ЛИТЕРАТУРЫ:
1. Positive Security. Уязвимости приложений финансовой отрасли. 2016. URL: https://www.ptsecurity.com/upload/ptru/analytics/Financial-Vulnerability-2016-rus.pdf (дата обращения: 22.02.2017)
2. OWASP. Education Track: What Developers Should Know on Web Application Security. URL: https://www.owasp.org/index.php/Education_Track:_What_Developers_Should_Know_on_Web_Application_Security (дата обращения: 23.02.2017)
3. Jøsang A. , Odegaard M, Oftedal E.: Cybersecurity Through Secure Software Development. 9th IFIP WG 11.8 World Conference, 2015
4. The National Initiative for Cybersecurity Education (NICE). URL: http://csrc.nist.gov/nice/ (дата обращения: 20.02.2017)
5. NIST. Creating a Patch and Vulnerability Management Program. URL: http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf
6. ГОСТ Р ИСО/МЭК 27034-1-2014. Информационная технология. Методы и средства обеспечения безопасности. Безопасность приложений. Часть 1. Обзор и общие понятия. М.: Стандартинформ, 2015. 64 с.
7. ISO/IEC TR 24772:2010. Information technology. Programming languages. Guidance to avoiding vulnerabilities in programming languages through language selection and use.
8. Microsoft Security Development Lifecycle. URL: https://www.microsoft.com/en-us/sdl/ (дата обращения: 15.02.2017)
9. Cisco Secure Development Lifecycle. URL: https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-secure-development-lifecycle.pdf (дата обращения: 15.02.2017)
10. Software Assurance Maturity Model. URL: http://www.opensamm.org/downloads/SAMM-1.0.pdf (дата обращения: 15.02.2017)
11. CLASP Security Principles. URL: https://www.owasp.org/index.php/CLASP_Security_Principles (дата обращения: 15.02.2017)
12. BSIMM Framwork. URL: https://www.bsimm.com/framework/ (дата обращения: 18.02.2017)
13. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhanov I. Synthesis of secure software development controls. In: SIN '15 Proceedings of the 8th International Conference on Security of Information and Networks, 2015
14. Burp Suite. URL: https://portswigger.net/burp/ (дата обращения: 21.02.2017)
15. OWASP Zed Attack Proxy Project. URL: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project (дата обращения: 21.02.2017)
16. Redmine. URL: https://www.redmine.org/ (дата обращения: 21.02.2017)
17. Fortify Static Code Analyzer. URL: http://www8.hp.com/ru/ru/software-solutions/static-code-analysis-sast/ (дата обращения: 21.02.2017)

REFERENCES:
[1] Positive Technologies. Vulnerabilities in Online Banking Systems - 2016. (In Russian. Available at: https://www.ptsecurity.com/upload/ptru/analytics/Financial-Vulnerability-2016-rus.pdf). (accessed 22.02.2017)
[2] OWASP. Education Track: What Developers Should Know on Web Application Security. (In Russian. Available at: https://www.owasp.org/index.php/Education_Track:_What_Developers_Should_Know_on_Web_Application_Security). (accessed 23.02.2017)
[3] Jøsang A. , Odegaard M, Oftedal E.: Cybersecurity Through Secure Software Development. In: 9th IFIP WG 11.8 World Conference, 2015
[4] The National Initiative for Cybersecurity Education (NICE). URL: http://csrc.nist.gov/nice/ (accessed: 20.02.2017)
[5] NIST. Creating a Patch and Vulnerability Management Program. URL: http://csrc.nist.gov/publications/nistpubs/800-40-Ver2/SP800-40v2.pdf
[6] GOST 27034-1-2014. Information technology. Security techniques. Application security. Part 1: Overview and concepts Moscow, Standartinform Publ., 2015. 64 p. (In Russian).
[7] ISO/IEC TR 24772:2010. Information technology. Programming languages. Guidance to avoiding vulnerabilities in programming languages through language selection and use.
[8] Microsoft Security Development Lifecycle. URL: https://www.microsoft.com/en-us/sdl/ (accessed: 15.02.2017)
[9] Cisco Secure Development Lifecycle. URL: https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-secure-development-lifecycle.pdf (accessed: 15.02.2017)
[10] Software Assurance Maturity Model. URL: http://www.opensamm.org/downloads/SAMM-1.0.pdf (accessed: 15.02.2017)
[11] CLASP Security Principles. URL: https://www.owasp.org/index.php/CLASP_Security_Principles (accessed: 15.02.2017)
[12] BSIMM Framwork. URL: https://www.bsimm.com/framework/ (accessed: 18.02.2017)
[13] Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhanov I. Synthesis of secure software development controls. In: SIN '15 Proceedings of the 8th International Conference on Security of Information and Networks, 2015
[14] Burp Suite. URL: https://portswigger.net/burp/ (accessed: 21.02.2017)
[15] OWASP Zed Attack Proxy Project. URL: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project (accessed: 21.02.2017)
[16] Redmine. URL: https://www.redmine.org/ (accessed: 21.02.2017)
[17] Fortify Static Code Analyzer. URL: http://www8.hp.com/ru/ru/software-solutions/static-code-analysis-sast/ (accessed: 21.02.2017)

Поступила в редакцию – 20 февраля июля 2017 г. Окончательный вариант – 21 мая 2017 г.
Received – February 20, 2017. The final version – May 21, 2017.