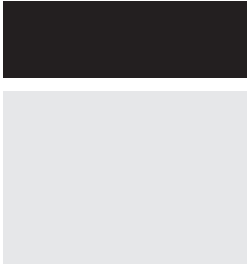


Я посвящаю эту книгу моему самому любимому человеку.

*Спасибо, что ты останавливал мир на время,
которого мне хватило, чтобы написать эту книгу.*

Я чувствую, что с твоей поддержкой могу все.



Содержание

От издательства	15
Об авторе	16
О технических редакторах	17
Благодарности	18
Вступительное слово	19
Введение	21
Часть I ОБЩИЕ РЕКОМЕНДАЦИИ	23
Глава 1 Введение в основы безопасности	24
Предполагайте, что все прочие системы и данные небезопасны	24
Триада CIA	25
Принцип наименьших привилегий	27
Безопасные умолчания и вымощенные дороги	29
Предполагайте возможность взлома, создавайте планы действий в чрезвычайных ситуациях	30
Нулевое доверие	31
Эшелонированная оборона	31
Безопасность цепочки поставок	32
Безопасность через неясность	32
Уменьшение поверхности атаки	33
Беспроblemная безопасность	33
После отказа закрыть, затем откатиться	34
Комплаенс, законы и нормативные акты	34
Каркасы безопасности	36
Учимся на ошибках и делимся уроками	38
Обратная совместимость (и сопутствующие ей потенциальные риски)	38
Моделирование угроз	39
Трудности обновления	39
Повторное тестирование исправлений новых дефектов	

безопасности	40
Упражнения	41
Глава 2 В начале пути	43
Применяйте безопасный жизненный цикл разработки ПО	43
Используйте современный каркас и все имеющиеся в нем средства безопасности	44
Проверка входных данных	45
Кодирование вывода	48
Примеры кодирования вывода	50
Контекст HTML.....	50
Контекст JavaScript	50
Параметризованные запросы и ORM	51
Аутентификация и идентификация	53
Авторизация и управление доступом.....	55
Модели управления доступом	56
Логические методы управления доступом (реализация).....	56
Управление сеансами.....	57
Управление секретами.....	58
Управление паролями	60
Безопасность связи (только криптография и HTTPS).....	62
Защита конфиденциальных данных.....	63
Заголовки, связанные с безопасностью.....	65
Новые функции заголовков, связанных с безопасностью.....	66
Заголовки запроса метаданных	66
Заголовок Content Security Policy.....	66
Strict-Dynamic	67
Trusted-Types.....	67
Ранее рассмотренные заголовки, связанные с безопасностью	67
Заголовок Content-Security-Policy.....	68
HTTP Strict-Transport-Security	68
X-Frame-Options	68
X-Content-Type-Options.....	68
Permissions-Policy.....	69
Expect-CT	69
Referrer-Policy.....	69
Расширение для фиксации открытого ключа HTTP (HPKP)	69
X-XSS-Protection	69
Еще несколько новых заголовков	69
Политика одного источника	70
COEP: Cross-Origin Embedder Policy.....	70
COOP: Cross-Origin Opener Policy.....	71
CORP: Cross-Origin Resource Policy	71
CORS: Cross-Origin Resource Sharing.....	72
CORB: Cross-Origin Read Blocking.....	72

Безопасные cookie.....	73
Обработка ошибок.....	74
Упражнения	75
Глава 3 Улучшение	77
Безопасность баз данных.....	78
Четыре взгляда на защиту баз данных.....	78
Управление файлами.....	81
Загрузка файлов на сервер	83
Ваш исходный код	84
Управление памятью (переполнение буфера, стека, строки и целого числа).....	85
Как избежать переполнения?	87
Сериализация и десериализация	88
Конфиденциальность (пользователя, гражданина, заказчика, работника).....	89
Ошибки.....	92
Протоколирование, мониторинг и уведомления.....	95
Закрытие при сбое	96
Блокировка ресурсов.....	97
Привлечение менеджеров паролей	98
Криптографическая практика.....	98
Строго и слабо типизированные языки	100
Строго типизированные языки	100
Слабо типизированные языки.....	101
Предметно-ориентированная разработка	102
Языки, безопасные относительно памяти	103
Упражнения	104
Глава 4 Достижение цели	105
Безопасное проектирование	106
Сколько безопасности (при проектировании) «достаточно»?	108
Управление зависимостями и безопасность цепочки поставок	110
Безопасность зависимостей.....	110
Проверяем, безопасно ли использовать зависимости	111
Безопасность цепочки поставок.....	111
Безопасные умолчания	114
Безопасные умолчания для пользователей.....	115
Безопасные умолчания для разработчиков.....	116
Код, удобный для чтения и аудита	118
Важные функции выполняются в надежных системах	121
Что такое «ненадежная» система?.....	121
Что такое «важные функции»?.....	122
Собираем все вместе	122
Черные и белые списки.....	123

Почему черные списки – это плохо?.....	123
Как создать белый список?.....	123
Безопасные конфигурации.....	124
Проверка имени хоста.....	125
Повторно используемый код	126
Безопасные системные вызовы	127
Смягчение последствий.....	128
Комментарии и другая документация.....	128
Комментарии	128
Документация	130
Запрос согласия пользователя	131
Проверки целостности, подписание кода и неизменяемые сборки ...	132
Неизменяемые сборки.....	134
Избегать грубой силы.....	135
Элементы безопасности.....	136
О расширенных привилегиях	137
Обслуживание безопасности.....	138
Оплата технического долга	139
Упражнения	141
Часть I – итоги	143
Контрольный список общих рекомендаций по безопасному кодированию	143
Часть II КОНКРЕТНЫЕ РЕКОМЕНДАЦИИ	151
Глава 5 Рекомендации, зависящие от технологии	152
Рекомендации по безопасности API	152
Рекомендации по безопасности мобильных приложений.....	159
Рекомендации по безопасности WebSocket.....	163
Рекомендации по бессерверным вычислениям.....	164
Рекомендации по безопасности IoT	165
Упражнения	167
Глава 6 Популярные языки программирования	169
JavaScript.....	169
HTML и CSS	174
Конкретно о HTML5.....	176
Python	178
SQL.....	180
Node.js.....	184
Java	187
Сериализация в Java	191
TypeScript	193
C#.....	194

PHP	198
C/C++	203
Заключение	206
Упражнения	207

Глава 7 Популярные каркасы..... 208

Веб и JavaScript	208
Express	208
React.js	210
Angular.....	213
jQuery.....	217
Vue.js	219
Другие каркасы и библиотеки.....	221
.NET (Core).....	221
Ruby on Rails	226
Spring и Spring Boot	232
Flask	234
Упражнения	238

Глава 8 Категории уязвимостей..... 239

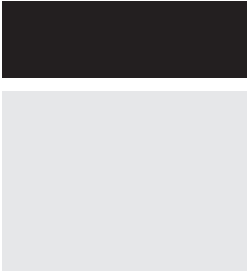
Просчеты проектирования (логические просчеты).....	240
Как такое возможно?.....	241
Риск.....	241
Предотвращение.....	242
Дефекты в коде (ошибки реализации).....	242
Как такое возможно?.....	243
Риск.....	243
Предотвращение.....	243
Переполнения и другие проблемы при работе с памятью.....	244
Переполнения	244
Чтение за границей буфера.....	245
Страничные отказы.....	245
Использование после освобождения.....	246
Неинициализированные переменные	246
Утечки памяти.....	246
Как такое возможно?.....	247
Риск.....	247
Предотвращение.....	247
Внедрение кода: проблемы интерпретатора и компилятора	249
Как такое возможно?.....	249
Риск.....	249
Предотвращение.....	250
Проблемы с входными данными.....	251
Как такое возможно?.....	251
Риск.....	252

Предотвращение	252
Проблемы аутентификации и идентификации	252
Как такое возможно?	252
Риск	253
Предотвращение	253
Проблемы авторизации и доступа	254
Как такое возможно?	254
Проблемы конфигурации и реализации	254
Как такое возможно?	255
Риск	255
Предотвращение	255
Мошеннические транзакции	256
Как такое возможно?	256
Риск	257
Предотвращение	257
Атаки воспроизведением	257
Как такое возможно?	258
Риск	258
Предотвращение	258
Пересечение границ доверия	259
Как такое возможно?	260
Риск	260
Предотвращение	260
Проблемы при работе с файлами	260
Как такое возможно?	260
Риск	261
Предотвращение	261
Проблемы при работе с объектами	261
Отличительные особенности ООП	262
Десериализация и другие проблемы при работе с объектами	264
Как такое возможно?	264
Риск	264
Предотвращение	264
Проблемы управления секретами	265
Как такое возможно?	266
Риск	266
Предотвращение	266
Состояния гонки и проблемы хронометража	267
Как такое возможно?	267
Риск	268
Предотвращение	268
Проблемы при работе с ресурсами	270
Как такое возможно?	270
Риск	271
Предотвращение	271

Попадание в неизвестное состояние	271
Как такое возможно?	272
Риск	272
Предотвращение	272
Упражнения	273
Часть II — итоги	274
Контрольный список рекомендаций по технологическим аспектам безопасного кодирования	274
Контрольный список рекомендаций по безопасному кодированию для языков и каркасов	275
Сводка уязвимостей, на которые следует обращать внимание	277
Часть III БЕЗОПАСНЫЙ ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ СИСТЕМЫ	279
Глава 9 Требования	280
Начало проекта: общий обзор действий по обеспечению безопасности	280
Составление графика и планирование проекта	281
Требования к безопасности	282
Упражнения	284
Глава 10 Проектирование	285
Моделирование угроз	286
Паттерны и концепции безопасного проектирования	288
Изображение архитектуры на доске	289
Изучение потоков данных	289
Пользовательские истории о безопасности	290
Упражнения	291
Глава 11 Кодирование	293
Обучение	293
Организации	295
Физические лица	296
Анализ кода	297
Статические анализаторы первого и второго поколения	297
Безопасные ограждения	299
Плагины IDE и другие наставления	299
Проверка безопасности зависимостей	300
Как решить, какие зависимости следует обновить или заменить?	301
Поиск секретов и управление ими	302
Динамическое тестирование (DAST)	303
Упражнения	305

Глава 12 Тестирование	306
Тестовое покрытие и хронометраж	307
Выбор между глубиной и широтой покрытия	308
Сканирование инфраструктуры	308
Производственная и менее ответственные среды	308
Охват	309
Хронометраж	309
Ручное тестирование	311
Автоматизированное тестирование	313
Фаззинг	315
Интерактивное тестирование приложений на безопасность (IAST)	315
Программы вознаграждения за найденные ошибки	316
Результаты тестирования	318
Действия по результатам тестирования	319
Заключительные мысли	321
Упражнения	321
Глава 13 Выпуск и развертывание	322
События безопасности в системе CI/CD	323
Прерывание сборки	324
Сканирование секретов	325
Статический анализ	325
Динамический анализ	325
Анализ состава программы	326
Линтинг	326
Сканеры инфраструктуры как кода	326
Обеспечение безопасности самого конвейера CI/CD	327
Поддержание целостности сборки	330
Одобрение выпуска группой безопасности	331
Упражнения	331
Глава 14 Сопровождение	333
Мониторинг, уведомление и наблюдаемость	334
Блокировка и экранирование	336
Брандмауэры веб-приложений (WAF)	337
Сети доставки содержимого (CDN)	337
Самозащита приложения на этапе выполнения (RASP)	338
Виртуальное латание	338
Шлюзы API	338
Замечание специально для исследователей данных	339
Непрерывное тестирование	340
Инциденты безопасности	341

Непрерывность бизнеса и планирование аварийного восстановления	343
Упражнения	345
Глава 15 Заключение	346
Хорошие привычки.....	346
Ваша обязанность	349
Сколько будет достаточно?.....	350
Безопасное использование искусственного интеллекта	352
Непрерывное обучение.....	354
Станьте ответственным за безопасность	356
Все на борт	357
Переход в группу безопасности.....	357
Устройство на работу в области безопасности в другой организации	358
Заключение	363
Часть III – итоги.....	366
Контрольный список действий, связанных с безопасностью, на каждом этапе SDLC	366
Приложение А. Ресурсы	369
Приложение В. Ответы к упражнениям.....	385
Предметный указатель	401



От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

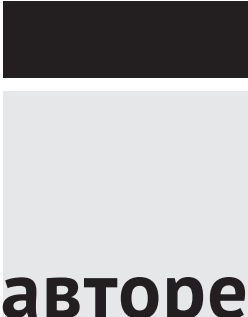
Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательства «ДМК Пресс» и Wiley очень серьезно относятся к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

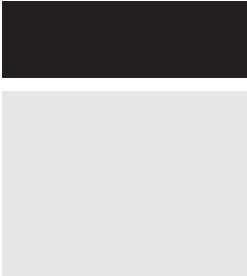
Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.



Об авторе

Таня Янца, она же SheHacksPurple, – автор бестселлеров «Alice and Bob Learn Secure Coding», «Alice and Bob Learn Application Security» и карточной игры AppSec Antics. За 28 лет карьеры в ИТ она стала лауреатом бесчисленных премий (включая звание пожизненного выдающегося члена OWASP и хакера года), выступала с докладами по всей планете и является плодовитым блогером. Таня обучила тысячи разработчиков программного обеспечения и специалистов по безопасности в своих онлайн-академиях (We Hack Purple и Semgrep Academy) и на практических семинарах. В рамках контртеррористической деятельности обеспечивала безопасность выборов 52-го генерал-губернатора Канады и обезопасила бесчисленное количество приложений. Таня Янца считается общепризнанным международным авторитетом по безопасности программного обеспечения.



О технических редакторах

Барре Дийкстра 25 лет занимается разработкой ПО на различных ролях – от практического программирования до руководства.

За это время он пришел к выводу о необходимости уделять больше внимания безопасности приложения, от кода до архитектуры и процесса. В настоящее время он часто оказывает поддержку клиентским командам по таким вопросам, как моделирование угроз, конфиденциальность и безопасность по определению, а также работает с группами обеспечения безопасности, занимающимися разработкой.

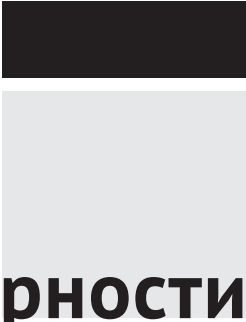
Помимо работы в сфере разработки ПО, Барре играет в CTF*, работает волонтером в Голландском институте обнаружения уязвимостей (Dutch Institute for Vulnerability Disclosure – DIVD), который принимает участие в организации одного из крупнейших соревнований по CTF, и несколько семестров преподавал на курсах подготовки бакалавров в области методов взлома и обратного проектирования.

Фабьен Ло Рикко – системный инженер с богатым 15-летним опытом в области конструирования ПО, системной инженерии и технической поддержки. Специализируется на обеспечении безопасности с упором на предотвращение потери данных (data loss prevention – DLP) и применяет свой практический опыт для защиты конфиденциальной информации.

Занимаясь на протяжении карьеры обслуживанием малых предприятий, он наработал разносторонний набор навыков, покрывающих весь спектр ИТ – от технической поддержки до проектирования архитектуры систем. Он активно выступает за ПО с открытым исходным кодом и применяет как проприетарные, так и открытые решения для проектирования надежных и экономичных систем, ориентированных на потребности бизнеса.

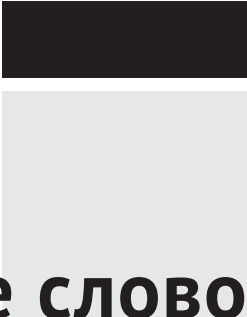
Франциска Пассинг – старший инженер-эксплуатационник в Дуолинго. Раньше работала в отделах инженерии внутренних платформ и обеспечения безопасности в таких компаниях, как Mapbox, Microsoft и др. Является опытным докладчиком, организатором технических сообществ и активным членом феминистского хакерского клуба Heart of Code. В настоящее время проживает в Берлине, где любит попивать кофе и участвовать в триатлонах. Связаться с ней можно через сайт <https://franka.tech>.

* Capture the Flag (захвати флаг) – командная игра по компьютерной безопасности. – Прим. перев.



Благодарности

Выражаю глубокую благодарность трем моим выдающимся техническим редакторам (Фабьену Ло Рикко, Франке Шмидт и Барре Дийкстра), чья непоколебимая поддержка, конструктивная критика и преданность были для меня стимулом на протяжении всего времени написания второй моей книги. Их воодушевление, настойчивость и стремление к совершенству не только повысили качество работы, но и вдохновили меня на достижение новых высот на моем писательском пути. Я в неоплатном долгу перед ними за их бесценный вклад и за то, что они находились со мной больше двух лет, сопровождая на всех подъемах и спусках процесса написания книги. Мне не хватает слов, чтобы в должной мере отблагодарить их за наставления, опыт и веру в проект.



Вступительное слово

С технической точки зрения сейчас отличное время быть инженером-программистом. Когда я была юным ботаном, зачитывалась научно-фантастическими романами и смотрела ранние эпизоды фильмов «Доктор Кто» и «Звездный путь: новое поколение», меня очаровывали технологии, демонстрируемые на экране. В том новом будущем у нас были приложения и приспособления для решения любой проблемы.

25 лет промелькнули быстро, и кажется, что мы сейчас живем в век необычайных изобретений. Программы проникли чуть ли не в каждую часть нашей жизни, от работы до свободного времени, а создаем их мы, инженеры.

Системы, которые мы пишем, объединяют нас и помогают сотрудничать и делиться информацией в невиданных ранее масштабах. Программы также помогают нам определять состояние здоровья, управлять роботами, которые помогают преодолевать неблагоприятные окружающие условия и даже ездят вместо нас по улицам наших городов. Однако при такой зависимости от программ и преимуществ, которые они предлагают, мы несем общую ответственность за то, что эти программы могут делать, и за их воздействие на людей, данные и системы.

В начале моей карьеры разработчика создаваемые мной системы (в области финансов и налогообложения) были не такими ответственными, как теперь. Эти системы обычно использовались только с 9 утра до 5 вечера с понедельника по пятницу, а если они вдруг переставали работать, то это, конечно, было неудобно, но компенсировалось многочисленными ручными системами.

Нынешние программы так глубоко вплетены в нашу жизнь, что последствия инцидентов и проблем значительно многообразнее и – в некоторых случаях – намного опаснее.

Именно поэтому настоящая книга и вся Танина работа по безопасному кодированию так важны. Мы, разработчики программного обеспечения, стоим на распутье. Мы обязаны следить за тем, чтобы наши программы и системы были высочайшего качества и отвечали самым строгим стандартам. Мы несем ответственность за баланс производительности, масштабируемости, удобства использования, наблюдаемости и доступности – всего того, благо-

даря чему наши приложения работают как задумано, когда это необходимо, обслуживая максимально возможное количество людей.

Самое время добавить сюда еще и безопасность.

В этой книге Таня постаралась сделать безопасное кодирование доступным разработчикам ПО на все случаи жизни и во всех программных экосистемах. Благодаря ее легким для понимания примерам и простому, прагматичному подходу даже самые занятые команды разработчиков смогут приступить к работе над безопасностью своего кода с учетом сложившегося у них цикла разработки, культуры и выбора языка. Изобилующая примерами и врезками, эта книга станет удобным справочником, который вы захотите вручить своей команде.

Если мы собираемся строить это чудесное пронизанное программами будущее вместе, то надеюсь, что сможем уделить время тому, чтобы сделать каждый следующий день немного безопаснее, чем предыдущий, а каждую строку кода будем рассматривать как возможность улучшить качество и безопасность своих систем.

Лаура Майн Белл,
генеральный директор и основатель компании SafeStack

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru