

Оглавление

Предисловие	14
Введение	18
Благодарности	20
О книге	22
Для кого эта книга	22
Структура книги	23
О коде.....	24
От издательства	25
Об авторах	26
Об иллюстрации на обложке	27

Часть I. Введение

Глава 1. Роль проектирования в безопасности	30
1.1. Безопасность как неотъемлемое свойство системы.....	32
1.1.1. Ограбление банка Öst-Götha, 1854 год.....	32
1.1.2. Элементы безопасности и безопасность в целом.....	34
1.1.3. Категории требований к безопасности: CIA-T.....	36
1.2. Что такое проектирование	37
1.3. Традиционный подход к безопасности ПО и его недостатки	40
1.3.1. Безопасность требует к себе отдельного внимания.....	42
1.3.2. Все должны быть специалистами по безопасности.....	43
1.3.3. Нужно знать обо всех уязвимостях, даже о неизвестных в данный момент	43
1.4. Обеспечение безопасности за счет проектирования.....	43
1.4.1. Делаем пользователя изначально защищенным.....	44
1.4.2. Преимущества подхода, основанного на проектировании	47
1.4.3. Сочетание разных подходов.....	50
1.5. Строки, XML и атака billion laughs	51
1.5.1. XML.....	51

1.5.2. Краткий обзор внутренних XML-сущностей.....	52
1.5.3. Атака Billion Laughs	53
1.5.4. Конфигурация XML-анализатора	53
1.5.5. Решение проблемы за счет проектирования	55
1.5.6. Применение операционных ограничений	58
1.5.7. Обеспечение глубокой безопасности	59
Резюме	60
Глава 2. Антракт: анти-«Гамлет».....	62
2.1. Книжный интернет-магазин с нарушением бизнес-целостности.....	65
2.1.1. Принцип работы журнала дебиторской задолженности.....	67
2.1.2. Как система складского учета отслеживает книги в магазине	68
2.1.3. Отправка антикниг.....	69
2.1.4. Системы верят в одну и ту же ложь	70
2.1.5. Самодельная скидка	71
2.2. Поверхностное моделирование	72
2.2.1. Откуда берутся поверхностные модели.....	74
2.2.2. Опасности, связанные с неявными концепциями.....	75
2.3. Глубокое моделирование.....	76
2.3.1. Как возникают глубокие модели	77
2.3.2. Пусть неявное становится явным	79
Резюме	80

Часть II. Основы

Глава 3. Основные концепции предметно-ориентированного проектирования	82
3.1. Модели как средства обеспечения более глубокого понимания.....	84
3.1.1. Модель — это упрощенная версия реальности.....	87
3.1.2. Модели должны быть строгими.....	90
3.1.3. Модели вбирают в себя глубокое понимание предметной области.....	94
3.1.4. Модель не создают, а выбирают	96
3.1.5. Модель формирует единый язык.....	98
3.2. Составные элементы модели	101
3.2.1. Сущности	102
3.2.2. Объекты-значения	106
3.2.3. Агрегаты.....	110
3.3. Ограниченные контексты	114
3.3.1. Семантика единого языка	114
3.3.2. Отношения между языком, моделью и ограниченным контекстом.....	115
3.3.3. Определение ограниченного контекста.....	116
3.4. Взаимодействие между контекстами.....	119
3.4.1. Использование одной модели в двух контекстах.....	119
3.4.2. Создание карты контекстов.....	121
Резюме	123

Глава 4. Концепции программирования, способствующие безопасности.....	125
4.1. Неизменяемость	126
4.1.1. Обыкновенный веб-магазин	126
4.2. Быстрое прекращение работы с использованием контрактов	134
4.2.1. Проверка предусловий для аргументов метода	137
4.2.2. Соблюдение инвариантов в конструкторах	139
4.2.3. Прекращение работы при обнаружении некорректного состояния	141
4.3. Проверка корректности	142
4.3.1. Проверка происхождения данных	144
4.3.2. Проверка размера данных	146
4.3.3. Проверка лексического содержимого данных.....	148
4.3.4. Проверка синтаксиса данных	150
4.3.5. Проверка семантики данных	152
Резюме	153
Глава 5. Доменные примитивы	155
5.1. Доменные примитивы и инварианты	156
5.1.1. Доменные примитивы — наименьшие составные элементы.....	156
5.1.2. Границы контекста определяют смысл.....	159
5.1.3. Создание собственной библиотеки доменных примитивов	162
5.1.4. Более надежные API на основе библиотеки доменных примитивов	162
5.1.5. Старайтесь не делать свою предметную область публично доступной.....	163
5.2. Объекты одноразового чтения.....	164
5.2.1. Обнаружение непреднамеренного использования данных	166
5.2.2. Предотвращение утечек в ходе развития кодовой базы	169
5.3. Опираясь на доменные примитивы.....	171
5.3.1. Риск загромождения методов сущностей.....	171
5.3.2. Оптимизация сущностей	174
5.3.3. Когда использовать доменные примитивы в сущностях.....	177
5.4. Анализ помеченных данных	178
Резюме	181
Глава 6. Обеспечение целостности состояния	182
6.1. Управление состоянием с помощью сущностей.....	183
6.2. Согласованность в момент создания.....	185
6.2.1. Опасность конструкторов, у которых нет аргументов	186
6.2.2. Фреймворки ORM и конструкторы без аргументов.....	188
6.2.3. Все обязательные поля в качестве аргументов конструктора	190
6.2.4. Создание объектов с использованием текучих интерфейсов	193
6.2.5. Соблюдение сложных ограничений в коде	195
6.2.6. Соблюдение сложных ограничений с помощью шаблона «Строитель»	197
6.2.7. Фреймворки ORM и сложные ограничения	201
6.2.8. В каких случаях использовать тот или иной метод создания	202

6.3. Целостность сущностей	202
6.3.1. Геттеры и сеттеры	203
6.3.2. Отказ от разделения изменяемых объектов	205
6.3.3. Обеспечение целостности коллекций.....	207
Резюме	210
Глава 7. Упрощение состояния	212
7.1. Частично неизменяемые сущности	215
7.2. Объекты состояния сущностей	216
7.2.1. Соблюдение правил о состоянии сущности	217
7.2.2. Реализация состояния сущности в виде отдельного объекта.....	221
7.3. Снимки сущностей.....	224
7.3.1. Сущности, представленные неизменяемыми объектами	224
7.3.2. Изменения состояния исходной сущности	227
7.3.3. Когда стоит использовать снимки	230
7.4. Эстафета сущностей.....	231
7.4.1. Разбиение диаграммы состояний на фазы.....	234
7.4.2. Когда стоит формировать эстафету сущностей	237
Резюме	239
Глава 8. Роль процесса доставки кода в безопасности	240
8.1. Использование конвейера доставки кода	241
8.2. Безопасное проектирование с использованием модульных тестов.....	242
8.2.1. Понимание правил предметной области	244
8.2.2. Проверка нормального поведения	245
8.2.3. Проверка граничного поведения.....	246
8.2.4. Тестирование с использованием недопустимого ввода.....	249
8.2.5. Тестирование экстремального ввода.....	252
8.3. Проверка переключателей функциональности	254
8.3.1. Опасность плохо спроектированных переключателей	254
8.3.2. Переключение функциональности как инструмент разработки	256
8.3.3. Укрощение переключателей	258
8.3.4. Комбинаторная сложность	262
8.3.5. Переключатели являются предметом аудита	263
8.4. Автоматизированные тесты безопасности	264
8.4.1. Тесты безопасности — это просто тесты	264
8.4.2. Использование тестов безопасности	265
8.4.3. Использование инфраструктуры как кода	266
8.4.4. Применение на практике	267
8.5. Тестирование доступности	267
8.5.1. Оценка операционного запаса	268
8.5.2. Эксплуатация правил предметной области.....	270
8.6. Проверка корректности конфигурации	271
8.6.1. Причины дефектов безопасности, связанных с конфигурацией.....	271

8.6.2. Автоматизированные тесты для подстраховки	273
8.6.3. Значения по умолчанию и их проверка	275
Резюме	277
Глава 9. Безопасная обработка сбоев.....	279
9.1. Использование исключений для обработки сбоев.....	280
9.1.1. Генерация исключений	281
9.1.2. Обработка исключений	284
9.1.3. Работа с полезным содержимым исключения	287
9.2. Обработка сбоев без использования исключений	289
9.2.1. Сбои не являются чем-то исключительным	290
9.2.2. Проектирование с учетом сбоев.....	291
9.3. Проектирование с расчетом на доступность	294
9.3.1. Устойчивость	294
9.3.2. Отзывчивость	295
9.3.3. Предохранители и тайм-ауты.....	296
9.3.4. Отсеки	298
9.4. Работа с некорректными данными.....	302
9.4.1. Не восстанавливайте данные перед проверкой корректности	303
9.4.2. Никогда не воспроизводите ввод дословно	305
Резюме	308
Глава 10. Преимущества облачного мышления	309
10.1. Концепции двенадцатифакторного приложения и облачной ориентированности..	310
10.2. Хранение конфигурации на уровне окружения	312
10.2.1. Не размещайте системную конфигурацию в коде.....	312
10.2.2. Никогда не храните конфиденциальные данные в файлах ресурсов	313
10.2.3. Хранение конфигурации на уровне окружения.....	315
10.3. Отдельные процессы.....	317
10.3.1. Развертывание и выполнение — это две разные вещи	318
10.3.2. Экземпляры обработки не хранят состояние	318
10.3.3. Преимущества с точки зрения безопасности	320
10.4. Не сохраняйте журнальные записи в файл	321
10.4.1. Конфиденциальность	322
10.4.2. Целостность.....	323
10.4.3. Доступность.....	323
10.4.4. Журналирование как услуга.....	324
10.5. Администраторские процессы.....	327
10.5.1. Риски безопасности, вызванные недостаточным вниманием к администраторским задачам	328
10.5.2. Администраторские задачи как полноправная часть системы.....	329
10.6. Обнаружение сервисов и балансировка нагрузки	331
10.6.1. Централизованная балансировка нагрузки.....	331
10.6.2. Балансировка нагрузки на стороне клиента	332
10.6.3. Адаптация к изменениям	333

10.7. Три составляющие корпоративной безопасности.....	334
10.7.1. Интенсивные изменения снижают риски	334
10.7.2. Ротация	335
10.7.3. Замена.....	337
10.7.4. Обновление	339
Резюме	340
Глава 11. Перерыв: страховой полис задаром	341
11.1. Продажа страховых полисов	342
11.2. Разделение сервисов.....	343
11.3. Новый тип платежей	345
11.4. Разбитая машина, запоздавший платеж и судебный иск.....	349
11.5. Что пошло не так?.....	352
11.6. Взгляд на общую картину происходящего	352
11.7. Замечание о микросервисной архитектуре	357
Резюме	358

Часть III. Применение основ на практике

Глава 12. Руководство по устаревшему коду	360
12.1. В какие участки старого кода следует внедрять доменные примитивы	361
12.2. Неоднозначные списки параметров	362
12.2.1. Прямолинейный подход	365
12.2.2. Аналитический подход.....	366
12.2.3. Подход с новым API	367
12.3. Сохранение в журнал непроверенных строк.....	368
12.3.1. Как непроверенные строки попадают в журнал	369
12.3.2. Обнаружение скрытой утечки данных	370
12.4. Защитные конструкции в коде.....	371
12.4.1. Код, который сам себе не доверяет	372
12.4.2. На помощь приходят контракты и доменные примитивы.....	374
12.4.3. Слишком небрежное использование типа Optional	376
12.5. Неправильное применение принципа DRY, когда во главе угла текст, а не идеи.....	377
12.5.1. Ложные срабатывания, к которым не нужно применять принцип DRY.....	378
12.5.2. Проблема объединения повторяющихся фрагментов кода	378
12.5.3. Правильное применение DRY	379
12.5.4. Ложноотрицательные срабатывания	379
12.6. Недостаточная проверка корректности в доменных типах.....	381
12.7. Тестирование на приемлемом уровне.....	382
12.8. Частичные доменные примитивы.....	384
12.8.1. Неявная контекстная валюта	385
12.8.2. Американский доллар — не то же самое, что словенский толар.....	386
12.8.3. Охват целостной концепции	387
Резюме	389

Глава 13. Руководство по микросервисам.....	390
13.1. Что такое микросервис.....	391
13.1.1. Независимые среды выполнения.....	392
13.1.2. Независимые обновления	392
13.1.3. Способность справляться со сбоями.....	393
13.2. Каждый сервис — это ограниченный контекст.....	393
13.2.1. Важность проектирования API	394
13.2.2. Разбиение монолита на части	397
13.2.3. Семантика и развивающиеся сервисы	397
13.3. Передача конфиденциальных данных между сервисами.....	398
13.3.1. CIA-T в микросервисной архитектуре	399
13.3.2. «Конфиденциальное» мышление	400
13.4. Ведение журнала в микросервисах.....	402
13.4.1. Целостность агрегированных журнальных данных	402
13.4.2. Отслеживаемость журнальных данных.....	404
13.4.3. Обеспечение конфиденциальности за счет предметно-ориентированного API для журналирования.....	406
Резюме	411
Глава 14. В заключение: не забывайте о безопасности!	413
14.1. Анализируйте код на предмет безопасности.....	414
14.1.1. Из чего должен состоять анализ кода на предмет безопасности	415
14.1.2. Кого привлекать к анализу кода на предмет безопасности.....	416
14.2. Следите за своим стеком технологий.....	417
14.2.1. Накопление информации	417
14.2.2. Расстановка приоритетов в работе	418
14.3. Проводите тестирование на проникновение	418
14.3.1. Проверка архитектурных решений.....	419
14.3.2. Учитесь на своих ошибках	420
14.3.3. Как часто следует проводить тестирование на проникновение	420
14.3.4. Использование программ bug bounty в качестве непрерывного тестирования на проникновение	421
14.4. Изучайте сферу безопасности	423
14.4.1. Базовое понимание безопасности должны иметь все	423
14.4.2. Безопасность как источник вдохновения.....	424
14.5. Выработайте процедуру на случай нарушения безопасности.....	425
14.5.1. Управление инцидентами	426
14.5.2. Решение проблем	427
14.5.3. Устойчивость, закон Вольффа и антихрупкость.....	428
Резюме	431