

Оглавление

Предисловие	14
Кому стоит прочитать эту книгу?	15
Разработчики, применяющие гибкие методы	15
Специалист по безопасности	16
Специалист по гибким методикам обеспечения безопасности.....	16
О структуре книги	16
Часть 1. Основы	17
Часть 2. Гибкая разработка и безопасность.....	17
Часть 3. Собираем все вместе	17
Графические выделения.....	18
Как с нами связаться	19
Благодарности	19
Глава 1. Начала безопасности	21
Не только для технарей.....	24
Безопасность и риск неразделимы.....	24
Уязвимость: вероятность и последствия	24
Все мы уязвимы	25
Не возможно, просто маловероятно	25
Измерение затрат	26
Риск можно свести к минимуму, но не устранить вовсе	27
Несовершенный мир – трудные решения	27
Знай своего врага	28
Враг найдется у каждого	28
Мотивы, ресурсы, доступ	30
Цели безопасности: защита данных, систем и людей	30
Понимание того, что мы пытаемся защитить.....	30
Конфиденциальность, целостность и доступность.....	30
Неотрицаемость.....	32
Соответствие нормативным требованиям, регулирование и стандарты безопасности.....	32
Типичные заблуждения и ошибки в области безопасности	33
Безопасность абсолютна	33
Безопасность – достижимое состояние	33
Безопасность статична	34
Для безопасности необходимо специальное [вставьте по своему усмотрению: пункт, устройство, бюджет]	34
Начнем, пожалуй	35
Глава 2. Элементы гибких методик	36
Сборочный конвейер	36
Автоматизированное тестирование	37

Непрерывная интеграция.....	41
Инфраструктура как код.....	42
Управление релизами.....	44
Визуальное прослеживание.....	46
Централизованная обратная связь.....	47
Хороший код – развернутый код.....	47
Работать быстро и безопасно.....	48
Глава 3. Революция в методах разработки – присоединяйтесь!	51
Гибкая разработка: взгляд с высоты.....	51
Scrum, самая популярная из гибких методик.....	54
Спринты и журналы пожеланий.....	54
Планерки.....	56
Циклы обратной связи в Scrum.....	57
Экстремальное программирование.....	58
Игра в планирование.....	58
Заказчик всегда рядом.....	59
Парное программирование.....	59
Разработка через тестирование.....	60
Метафора системы.....	61
Канбан.....	61
Канбан-доска: сделать работу видимой.....	63
Постоянная обратная связь.....	63
Непрерывное улучшение.....	64
Бережливая разработка.....	64
Гибкие методы в целом.....	66
А как насчет DevOps?.....	68
Гибкие методики и безопасность.....	71
Глава 4. Работа с существующим жизненным циклом гибкой разработки	73
Традиционные модели безопасности приложения.....	73
Ритуалы на каждой итерации.....	76
Инструменты, встроенные в жизненный цикл.....	78
Деятельность до начала итераций.....	79
Инструменты планирования и обнаружения.....	80
Деятельность после итерации.....	80
Инструментальные средства в помощь команде.....	81
Инструменты проверки соответствия нормативным требованиям и аудита.....	82
Задание контрольного уровня безопасности.....	82
А что будет при масштабировании?.....	83
Создание содействующих групп безопасности.....	83
Создание инструментов, которыми будут пользоваться.....	84
Методы документирования системы безопасности.....	85
Сухой остаток.....	86

Глава 5. Безопасность и требования	87
Учет безопасности в требованиях.....	87
Гибкие требования: рассказывание историй.....	89
Как выглядят истории?.....	89
Условия удовлетворенности.....	90
Учет историй и управление ими: журнал пожеланий.....	91
Отношение к дефектам.....	92
Включение вопросов безопасности в требования.....	92
Истории, касающиеся безопасности.....	93
Конфиденциальность, мошенничество, соответствие нормативным требованиям и шифрование.....	97
Истории, касающиеся безопасности, с точки зрения SAFECode.....	99
Персоны и антиперсоны безопасности.....	101
Истории противника: надеваем черную шляпу.....	103
Написание историй противника.....	105
Деревья атак.....	107
Построение дерева атак.....	108
Сопровождение и использование деревьев атак.....	109
Требования к инфраструктуре и эксплуатации.....	110
Сухой остаток.....	115
Глава 6. Гибкое управление уязвимостями	116
Сканирование на уязвимости и применение исправлений.....	116
Сначала поймите, что сканировать.....	117
Затем решите, как сканировать и с какой частотой.....	118
Учет уязвимостей.....	119
Управление уязвимостями.....	120
Как относиться к критическим уязвимостям.....	124
Обеспечение безопасности цепочки поставок программного обеспечения.....	125
Уязвимости в контейнерах.....	127
Лучше меньше, да лучше.....	127
Как устранить уязвимости по-гибкому.....	128
Безопасность через тестирование.....	130
Нулевая терпимость к дефектам.....	131
Коллективное владение кодом.....	132
Спринты безопасности, спринты укрепления и хакатоны.....	133
Долг безопасности и его оплата.....	135
Сухой остаток.....	137
Глава 7. Риск для гибких команд	139
Безопасники говорят «нет».....	139
Осознание рисков и управление рисками.....	140
Риски и угрозы.....	142
Отношение к риску.....	143

Делать риски видимыми	144
Принятие и передача рисков	145
Изменение контекста рисков	146
Управление рисками в гибких методиках и DevOps	148
Скорость поставки	149
Инкрементное проектирование и рефакторинг	150
Самоорганизующиеся автономные команды	151
Автоматизация	152
Гибкое смягчение риска	152
Отношение к рискам безопасности в гибких методиках и DevOps	155
Сухой остаток	157
Глава 8. Оценка угроз и осмысление атак	159
Осмысление угроз: паранойя и реальность	159
Понимание природы злоумышленников	160
Архетипы злоумышленников	161
Угрозы и цели атаки	164
Разведка угроз	165
Оценка угроз	168
Поверхность атаки вашей системы	169
Картирование поверхности атаки приложения	170
Управление поверхностью атаки приложения	171
Гибкое моделирование угроз	173
Доверие и границы доверия	173
Построение модели угроз	176
«Достаточно хорошо» – и достаточно	176
Думать как противник	179
STRIDE – структурная модель для лучшего понимания противника	180
Инкрементное моделирование угроз и оценка рисков	181
Оценка рисков в самом начале	181
Пересмотр угроз при изменении проекта	182
Получение выгоды от моделирования угроз	183
Типичные векторы атак	185
Сухой остаток	186
Глава 9. Построение безопасных и удобных для пользования систем	188
Проектируйте с защитой от компрометации	188
Безопасность и удобство пользования	189
Технические средства контроля	190
Сдерживающие средства контроля	190
Средства противодействия	191
Защитные средства контроля	191
Детекторные средства контроля	192
Компенсационные средства контроля	192

Архитектура безопасности.....	193
Безопасность без периметра.....	193
Предполагайте, что система скомпрометирована.....	196
Сложность и безопасность.....	197
Сухой остаток.....	199
Глава 10. Инспекция кода в интересах безопасности	200
Зачем нужна инспекция кода?	200
Типы инспекций кода	202
Формальные инспекции.....	202
Метод утенка, или Проверка за столом	202
Парное программирование (и программирование толпой)	203
Дружественная проверка	204
Аудит кода	204
Автоматизированная инспекция кода.....	205
Какой подход к инспекции оптимален для вашей команды?.....	205
Когда следует инспектировать код?.....	206
До фиксации изменений.....	206
Контрольно-пропускные проверки перед релизом	207
Посмертное расследование.....	207
Как проводить инспекцию кода.....	208
Применяйте наставление по кодированию	208
Контрольные списки для инспекции кода.....	209
Не делайте этих ошибок	210
Инспектируйте код небольшими порциями	211
Какой код следует инспектировать?.....	212
Кто должен инспектировать код?.....	214
Сколько должно быть инспекторов?.....	215
Каким опытом должны обладать инспекторы?	216
Автоматизированная инспекция кода.....	217
Разные инструменты находят разные проблемы	219
Какие инструменты для чего подходят.....	221
Приучение разработчиков к автоматизированным инспекциям кода	224
Сканирование в режиме самообслуживания.....	226
Инспекция инфраструктурного кода	228
Недостатки и ограничения инспекции кода	229
Для инспекции нужно время.....	230
Разобраться в чужом коде трудно.....	231
Искать уязвимости еще труднее	231
Внедрение инспекций кода на безопасность	233
Опирайтесь на то, что команда делает или должна делать	233
Рефакторинг: поддержание простоты и безопасности кода	235
Базовые вещи – вот путь к безопасному и надежному коду.....	236
Инспекция функций и средств контроля, относящихся к безопасности.....	238

Инспекция кода на предмет угроз от инсайдеров.....	239
Сухой остаток.....	241
Глава 11. Гибкое тестирование безопасности	244
Как производится тестирование в гибких методиках?	244
Кто допускает ошибки, тот побежден.....	246
Пирамида гибкого тестирования.....	247
Автономное тестирование и TDD.....	249
Последствия автономного тестирования для безопасности системы.....	250
Нам не по пути успеха	251
Тестирование на уровне служб и средства BDD.....	253
GauntIt («придирайся к своему коду»).....	253
BDD-Security.....	254
Заглянем под капот.....	254
Приемочное тестирование.....	256
Функциональное тестирование и сканирование безопасности.....	256
Краткое пособие по ZAP.....	257
ZAP в конвейере непрерывной интеграции	259
Совместное использование BDD-Security и ZAP.....	260
Трудности, возникающие при сканировании приложений.....	262
Тестирование инфраструктуры.....	265
Проверка правил оформления.....	267
Автономное тестирование.....	267
Приемочное тестирование.....	267
Создание автоматизированного конвейера сборки и тестирования.....	269
Ночная сборка.....	270
Непрерывная интеграция.....	270
Непрерывная поставка и непрерывное развертывание.....	271
Экстренное тестирование и инспекция	272
Передача в эксплуатацию	273
Рекомендации по созданию успешного автоматизированного конвейера	274
Место тестирования безопасности в конвейере.....	274
Место ручного тестирования в гибких методиках	276
Как добиться, чтобы тестирование безопасности работало в гибких методиках и DevOps?	278
Сухой остаток.....	280
Глава 12. Внешние инспекции, тестирование и рекомендации.....	282
Почему нужны внешние инспекции?.....	283
Оценка уязвимости	286
Тестирование на проникновение	287
Команда красных	291
Вознаграждение за обнаружение ошибок.....	293
Как работает программа вознаграждения.....	293

Подготовка к программе вознаграждения за найденные ошибки	294
А вы уверены, что хотите запустить программу вознаграждения?	299
Инспекция конфигурации	303
Аудит безопасности кода	303
Криптографический аудит	304
Выбор сторонней компании	306
Опыт работы с продуктами и организациями, похожими на ваши	307
Активная исследовательская работа и повышение квалификации	307
Встречи с техническими специалистами	308
Оценка результатов оплаченной работы	308
Не тратьте чужое время попусту	309
Проверка найденных проблем	309
Настаивайте на устраивающей вас форме результатов	310
Интерпретируйте результаты в контексте	310
Подключайте технических специалистов	310
Измеряйте улучшение со временем	310
Храните сведения о состоявшихся инспекциях и ретроспективном анализе и делитесь результатами	311
Распределяйте устранение проблем между командами, чтобы способствовать передаче знаний	311
Время от времени ротлируйте оценщиков или меняйте местами тестировщиков	311
Сухой остаток	312
Глава 13. Эксплуатация и безопасность	314
Укрепление системы: настройка безопасных систем	315
Нормативно-правовые требования к укреплению	317
Стандарты и рекомендации, относящиеся к укреплению	318
Проблемы, возникающие при укреплении	319
Автоматизированное сканирование на соответствие нормативным требованиям	321
Подходы к построению укрепленных систем	322
Автоматизированные шаблоны укрепления	324
Сеть как код	325
Мониторинг и обнаружение вторжений	327
Мониторинг с целью организации обратной связи	328
Использование мониторинга приложений в интересах безопасности	328
Аудит и протоколирование	330
Проактивное и реактивное обнаружение	333
Обнаружение ошибок во время выполнения	334
Оборона во время выполнения	336
Обеспечение безопасности в облаке	336
RASP	337
Реакция на инциденты: подготовка к взлому	340
Тренируйтесь: учения и команда красных	340

Посмертный анализ без поисков виновного: обучение на инцидентах безопасности.....	342
Защита сборочного конвейера	344
Укрепление инфраструктуры сборки.....	346
Выяснение того, что происходит в облаке.....	346
Укрепление инструментов непрерывной интеграции и поставки.....	347
Ограничение доступа к диспетчерам конфигурации	349
Защита ключей и секретов.....	349
Ограничение доступа к репозиториям	349
Безопасный чат	350
Просмотр журналов	351
Использование серверов-фениксов для сборки и тестирования.....	351
Мониторинг систем сборки и тестирования	352
Шшш... секреты должны храниться в секрете.....	352
Сухой остаток.....	355
Глава 14. Соответствие нормативным требованиям	357
Соответствие нормативным требованиям и безопасность.....	358
Различные подходы к законодательному регулированию.....	361
PCI DSS: подход на основе правил.....	362
Надзор за целостностью и соблюдением требований: подход на основе результатов	366
Какой подход лучше?.....	367
Управление рисками и соответствие нормативным требованиям.....	367
Прослеживаемость изменений	369
Конфиденциальность данных.....	370
Как соответствовать нормативным требованиям, сохраняя гибкость.....	372
Истории о соответствии и соответствие в историях.....	373
Больше кода, меньше писанины.....	374
Прослеживаемость и гарантии непрерывной поставки	376
Управление изменениями при непрерывной поставке.....	379
Разделение обязанностей	381
Встраивание соответствия нормативным требованиям в корпоративную культуру	383
Как доставить удовольствие аудитору	384
Как быть, когда аудиторы недовольны	386
Сертификация и аттестация.....	387
Непрерывное соответствие и взломы.....	387
Сертификация не означает, что вы в безопасности.....	388
Сухой остаток.....	388
Глава 15. Культура безопасности	390
Важность культуры безопасности.....	391
Определение «культуры».....	391
Тяни, а не толкай.....	391

Выстраивание культуры безопасности	392
Принципы эффективной безопасности	393
Содействуй, а не блокируй	395
Прозрачная безопасность	399
Не ищите виноватых	401
Масштабировать безопасность, усиливать фланги	405
Кто – не менее важно, чем как	407
Продвижение безопасности	408
Эргобезопасность	410
Информационные панели	412
Сухой остаток	417
Глава 16. Что такое гибкая безопасность?	418
История Лауры	418
Не инженер, а хакер	418
Твое дитя – уродец, и ты должен чувствовать себя виноватым	419
Поменьше говори, побольше слушай	420
Давайте двигаться быстрее	420
Создание круга поклонников и друзей	421
Мы невелички, но нас много	421
История Джима	422
Вы можете вырастить собственных экспертов по безопасности	422
Выбирайте людей, а не инструменты	424
Безопасность должна начинаться с качества	425
Соответствие нормативным требованиям может стать повседневной практикой ..	426
История Майкла	426
Знания о безопасности распределены неравномерно	429
Практическим специалистам нужно периодически проходить повышение квалификации	430
Аккредитация и гарантии отмирают	430
Безопасность должна содействовать делу	431
История Рича	431
Первый раз бесплатно	432
А это может быть не просто хобби?	433
Прозрение	433
С компьютерами трудно, с людьми еще труднее	434
И вот мы тут	435
Сведения об авторах	436
Об иллюстрации на обложке	438
Предметный указатель	439

Предисловие

Программы правят миром. Разработчики стали новыми «делателями королей». Благодаря интернету вещей компьютер скоро появится в каждой лампочке.

Все это означает, что скоро программы настолько глубоко проникнут в нашу жизнь, что для большинства людей до ближайшего компьютера будет буквально рукой подать, и постоянное взаимодействие с предметами и окружением, управляемыми компьютерами, войдет у нас в привычку.

Но в таком мире нас подстерегают опасности. В старом добром мире безопасность всерьез рассматривалась только в банковских и правительственных системах. Однако повсеместное распространение компьютеров повышает потенциальную выгоду от их непропорционального использования, что, в свою очередь, создает стимулы для такого использования и, стало быть, повышает риски, с которыми сталкиваются системы.

В большинстве организаций стремительно принимают на вооружение гибкие (agile) методики разработки. Они позволяют своевременно реагировать на изменение условий и значительно снижать стоимость разработки, поэтому задают стандарт, который, как мы ожидаем, будет распространяться все шире, и в конечном итоге большая часть программ будет создаваться с помощью гибких методик.

Однако исторически безопасность и гибкие методики никогда не дружили между собой.

Специалисты по безопасности по горло заняты уже упомянутыми системами для государственных учреждений, банков и электронной коммерции – их архитектурой, тестированием и защитой, – и все это перед лицом непрерывно совершенствующихся угроз. То, что часто представляют самым интересным и захватывающим в области безопасности, то, о чем пишут в технических блогах и говорят в ночных новостях, делается командами профессиональных хакеров, специализирующихся на исследовании уязвимостей, разработке эксплойтов и ошеломительных взломах.

Наверняка вы сможете назвать несколько уязвимостей, бывших на слуху в последнее время: Heartbleed, Logjam или Shellshock (или еще каких-нибудь, с которыми на приведи бог столкнуться), или припомнить названия команд, которым удалось «разлочить» последние модели

устройств на основе iPhone и Android. Но когда в последний раз вы слышали о новом средстве или методике защиты – с запоминающимся, интересным для СМИ названием? А знаете ли вы имя создателя этой методики?

Профессионалы в области безопасности отстают в понимании и опыте применения гибких методик, и этот разрыв несет серьезные риски для нашей отрасли.

С другой стороны, гибкие команды отбросили стесняющие оковы прошлого. Нет больше детальных технических требований, нет системного моделирования, не стало каскадной модели (модели «водопада») с ее традиционными этапами и контролем их выполнения. Беда в том, что гибкие команды вместе с водой выплеснули и ребенка. Старые методики, пусть даже иногда медленные и негибкие, доказали свою ценность. Они создавались не без причины, а отказавшись от них, гибкие команды рискуют забыть, в чем состояла эта ценность, и выбросить все достижения на свалку.

Поэтому гибкие команды редко относятся к безопасности, как должно. Некоторые гибкие практики позволяют получить чуть более безопасные системы, но зачастую это благоприятный побочный эффект, а не цель. И лишь очень немногие гибкие команды понимают угрозы, с которыми может столкнуться система, а большинство не понимает рисков, не отслеживает их и не делает ничего, чтобы управлять ими. Они даже слабо представляют, кто вообще атакует плоды их труда.

Кому стоит прочитать эту книгу?

Мы не знаем, кто вы: руководитель гибкой команды или разработчик, желающий больше узнать о безопасности. Быть может, вы – «безопасник», который только что узнал о существовании доселе неведомой ему группы разработчиков и хотел бы разобраться, чем она занята.

При написании этой книги мы имели в виду три потенциальные группы читателей.

Разработчики, применяющие гибкие методы

Вы жизни не мыслите без гибкой разработки. Scrum и кайдзен – для вас не пустые слова, разработка через тестирование вошла в плоть и кровь. Не важно, какую конкретную роль вы играете – Scrum-мастер, разработчик, тестировщик, тренер, владелец продукта или представитель заказчика, – вы владеете гибкими практиками и понимаете, в чем их ценность.

Эта книга поможет вам разобраться в том, что такое безопасность, какие существуют угрозы и на каком языке специалисты-безопасники описывают, что происходит. Мы научим вас моделировать угрозы, измерять степень риска, создавать ПО, помня о безопасности, безопасно устанавливать ПО и оценивать, какие проблемы (в плане безопасности) могут возникать в процессе его эксплуатации.

Специалист по безопасности

Если вы риск-менеджер, специалист по обеспечению целостности и безопасности информации (information assurance) или по анализу операционной безопасности, то что такое безопасность, вы понимаете. Наверное, вы с осторожностью подходите к использованию онлайн-услуг, постоянно думаете об угрозах, рисках и формах их проявления, возможно, даже сами находили новые уязвимости и разрабатывали эксплойты для них.

Эта книга поможет вам понять, как работают гибкие команды и что, наконец, такое спринты и истории, о которых они все время толкуют. Вы научитесь видеть порядок в хаосе, и это поможет взаимодействовать с командой и влиять на нее. Эта книга покажет, где следует вмешаться или внести свой вклад, чтобы он оказался наиболее ценным для команды и дал наилучший эффект.

Специалист по гибким методикам обеспечения безопасности

Вы знаете все – от рисков до спринтов. Будь вы разработчиком инструментальных средств, который хочет помочь команде правильно подойти к безопасности, или консультантом, оказывающим команде платные услуги, эта книга и для вас тоже. Главное в ней – уяснить то, что авторы называют возрастанием доли правильной практики. Эта книга поможет вам узнать, кто работает с вами по соседству, а также составить представление об идеях, размышлениях и концепциях, которые, как мы видим, прорастают в организациях, работающих над этой проблематикой. Мы дадим широкий обзор предмета, чтобы вы смогли решить, чем заняться или что изучить дальше.

О структуре книги

Книгу можно читать от корки до корки, по одной главе за раз. Собственно, так мы и рекомендуем поступить; мы приложили немало трудов и

надеемся, что в каждой главе любой читатель найдет что-то полезное для себя, пусть даже это будут просто наши незатейливые шутки и забавные рассказы!

Но, честно говоря, мы полагаем, что некоторые главы окажутся для вас полезнее, чем другие.

Книга разбита на три большие части.

Часть 1. Основы

Гибкая разработка и безопасность – очень широкие дисциплины, и мы не знаем, что вам уже известно. А если вы специализируетесь в одной области, то вполне возможно, что знаний и опыта из другой вам недостает.

Если вы специалист по гибкой разработке, то рекомендуем сначала прочитать *главу 1 «Начала безопасности»*, из которой вы получите базовые сведения о безопасности.

Если раньше вы вообще не занимались гибкой разработкой или делаете первые шаги на этом пути, то, прежде чем переходить к введению в эту тему, рекомендуем прочитать *главу 2 «Элементы гибких методик»*.

В *главе 3 «Революция в методах разработки – присоединяйтесь!»* описывается история гибкой разработки программного обеспечения и различные подходы к ней. Эта глава будет особенно интересна специалистам по безопасности и людям, еще не имеющим практического опыта.

Часть 2. Гибкая разработка и безопасность

Затем мы рекомендуем всем без исключения прочитать *главу 4 «Работа с существующим гибким жизненным циклом»*.

В этой главе производится попытка соединить рассматриваемую нами практическую безопасность с реальным жизненным циклом гибкой разработки и объяснить, как они сочетаются.

В *главах 5–7* разбирается управление требованиями и уязвимостью, а также управление рисками. Это общие принципы, лежащие в основе управления продуктом и планирования разработки.

В *главах 8–13* рассматриваются различные аспекты жизненного цикла разработки безопасного программного обеспечения: оценка угроз, инспекция кода, тестирование и операционная безопасность.

Часть 3. Собираем все вместе

В *главе 14* рассматривается соответствие нормативным требованиям и их учет в средах гибкой разработки и DevOps.

Глава 15 посвящена культурным аспектам безопасности. Да, вы могли бы реализовать все описанные в книге практики, и в предшествующих главах описаны разнообразные средства, позволяющие затвердить новые подходы. Но гибкие методики – это в первую очередь о людях, и то же самое верно в отношении эффективных путей обеспечения безопасности: безопасность – это внутренне осознанное изменение культуры поведения, и в этой главе мы приведем примеры приемов, которые оказались эффективны на практике.

Чтобы изменить отношение компании к безопасности, необходимы взаимная поддержка и уважение безопасников и разработчиков. Чтобы создать безопасный продукт, требуется их тесное взаимодействие. Этого невозможно добиться только путем внедрения набора средств и практик, изменения должны пронизывать всю организацию.

Наконец, в главе 16 рассматривается вопрос о том, что разные люди понимают под «гибкой безопасностью», и подводится итог опыту авторов: что у них получалось и что не получалось при попытке создать команды, сочетающие одновременно гибкость и безопасность.

Графические выделения

В книге применяются следующие графические выделения:

Курсив

Новые термины, URL-адреса, адреса электронной почты, имена и расширения имен файлов.

Моноширинный шрифт

Листинги программ, а также элементы кода в основном тексте: имена переменных и функций, базы данных, типы данных, переменные окружения, предложения и ключевые слова языка. Знак ↵ в конце строки кода означает, что код продолжается на следующей строке.

Моноширинный полужирный шрифт

Команды и другой текст, который пользователь должен вводить буквально.

Моноширинный курсив

Текст, вместо которого следует подставить значения, заданные пользователем или определяемые контекстом.



Так обозначается совет или рекомендация.



Так обозначается примечание общего характера.



Так обозначается предупреждение или предостережение.

Как с нами связаться

Вопросы и замечания по поводу этой книги отправляйте в издательство:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (в США и Канаде)
707-829-0515 (международный или местный)
707-829-0104 (факс)

Для этой книги создана веб-страница, на которой публикуются сведения о замеченных опечатках, примеры и разного рода дополнительная информация. Адрес страницы <http://bit.ly/agile-application-security>.

Замечания и вопросы технического характера следует отправлять по адресу bookquestions@oreilly.com.

Дополнительную информацию о наших книгах, конференциях и новостях вы можете найти на нашем сайте по адресу <http://www.oreilly.com>.

Читайте нас на Facebook: <http://facebook.com/oreilly>.

Следите за нашей лентой в Twitter: <http://twitter.com/oreillymedia>.

Смотрите нас на YouTube: <http://www.youtube.com/oreillymedia>.

Благодарности

Прежде всего мы благодарны нашим замечательным редакторам: Кортни Аллен, Вирджинии Уилсон и Нэн Барбер. Мы не смогли бы довести это дело до конца без вас и всего коллектива издательства O'Reilly.

Мы также выражаем признательность техническим рецензентам за их терпение и полезные советы: Бену Аллену (Ben Allen), Джеффу Кратцу (Geoff Kratz), Питу Макбрину (Pete McBreen), Келли Шортридж (Kelly Shortridge) и Ненаду Стояновски (Nenad Stojanovski).

И наконец, спасибо нашим друзьям и семьям, выдержавшим *еще один* безумный проект.

Глава 1

Начала безопасности

Итак, что же такое безопасность?

Обманчиво простой вопрос, на который куда как сложно ответить.

Начиная путешествие в мир безопасности, трудно не то что разобратся, но хотя бы понять, куда смотреть первым делом. В новостях об успешных *взломах* рисуют картину злоумышленника типа Нео, который располагает чуть ли не безграничным арсеналом для проведения сложнейших атак. При такой точке зрения обеспечение безопасности может показаться безнадежным делом, выходящим за рамки человеческих возможностей.

Да, действительно, безопасность – сложная, постоянно изменяющаяся дисциплина, но верно и то, что существует ряд довольно простых базовых принципов, поняв которые, будет куда легче систематизировать знания, приобретенные впоследствии. Рассматривайте обеспечение безопасности как движение вперед, а не как конечную цель – движение, начинающееся с небольшого числа фундаментальных понятий, отталиваясь от которых, можно постепенно строить все здание.

Поэтому важно, чтобы все мы, независимо от прежнего опыта, первым делом уяснили некоторые основополагающие принципы. Мы также рассмотрим традиционные подходы к безопасности и объясним, почему они перестали быть эффективными теперь, когда гибкие методы разработки стали вездесущими.

С точки зрения команд разработчиков, под безопасностью понимается информационная безопасность (а не физическая безопасность, воплощенная в дверях и стенах, а также в организации охраны, например процедурах досмотра персонала). Информационная безопасность включает практические приемы и процедуры в начале работы над проектом, в процессе реализации системы и в ходе ее эксплуатации.



Хотя в этой книге мы будем говорить главным образом *об информационной безопасности*, для краткости будем употреблять просто слово *безопасность*. Если понадобится упомянуть о каком-то другом аспекте безопасности, например физическом, то это будет оговорено явно.

Инженеры часто обсуждают, какую технологию выбрать для систем и сред. Но вопросы безопасности вынуждают нас выходить за рамки технологий. Быть может, безопасность лучше всего представлять как область на пересечении технологии с людьми, которые эту технологию используют для повседневных надобностей (см. рис. 1.1).

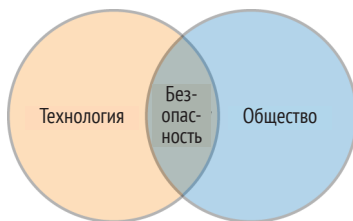


Рис. 1.1. Чем меньше общество зависит от технологии, тем меньше потребность в безопасности

Что можно сказать, глядя на эту картинку? Она показывает, что безопасность – нечто большее, чем просто технология, и, по самому своему определению, *должна* включать людей.

Людам не нужна технология, чтобы совершать дурные поступки и обманывать друг друга; этим они занимались задолго до того, как в нашу жизнь вошли компьютеры, для этого даже слово есть – *преступление*. Тысячи лет люди совершенствовались в искусстве лгать, обманывать и красть на пользу себе и своему сообществу. Но когда человек начинает взаимодействовать с технологией, это превращается в многообещающую комбинацию мотивов, целей и возможностей. В таких случаях некоторые мотивированные группы людей проводят согласованные действия в обход технологических ограничений во имя достижения конечной цели – вполне в духе человеческой природы. Именно такую деятельность и призвана предотвратить безопасность.

Следует, однако, отметить, что технический прогресс создал условия для расширения братства людей, способных на такие преступления: как благодаря доступности инструкций, так и вследствие появления всемирных сервисов, к которым может попытаться получить доступ мотивированный преступник. При наличии Интернета, мировых си-

стем связи и других достижений прогресса атаковать вас намного проще, чем раньше, притом что для преступника риск попасться гораздо ниже. Интернет и сопутствующие ему технологии сделали мир совсем маленьким, а асимметрия при этом стала даже более разительной – затраты снизились, прибыль возросла, а шансы быть пойманными резко уменьшились. В этом новом мире географическое расстояние до вождельной богатой добычи для атакующего свелось к нулю, но действует прежняя юридическая система, основанная на межгосударственных соглашениях и процедурах межюрисдикционных расследований и экстрадиций. И это мы еще не говорим о различиях принятых в разных странах определений существа компьютерного преступления. Технологии и Интернет также помогают взломщику избежать идентификации: вам больше не нужно явиться в банк, чтобы ограбить его, – вы можете находиться на другом конце света.



Замечание по поводу терминологии

При обсуждении небезопасности мы сознательно употребляем фразу «в обход», чтобы избежать неявных моральных суждений.

Чем больше в нашей жизни технологий, тем больше у нас возможностей использовать их во благо. Но у этой медали есть и обратная сторона: чем выше зависимость общества от технологий, тем больше возможностей и стимулов использовать их во зло и тем выше доход от этого. Чем сильнее мы зависим от технологии, тем больше потребность в том, чтобы она была стабильна, защищена и всегда доступна. Если стабильность и безопасность оказываются под вопросом, страдает и бизнес, и общество. Тот же рисунок, что и выше, иллюстрирует взаимозависимость между уровнем распространения технологий в обществе и необходимостью обеспечить безопасность во имя достижения стабильности и защищенности (рис. 1.2).

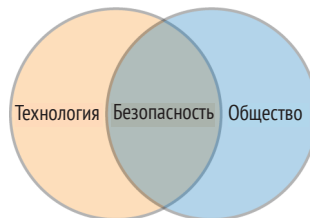


Рис. 1.2. Чем сильнее зависимость от технологии, тем выше потребность в безопасности и серьезнее последствия ее отсутствия

По мере проникновения технологии в ткань общества все большую важность приобретают подходы к осмыслению ее безопасности.

Фундаментальный недостаток классических подходов к информационной безопасности – непонимание того, что люди не менее важны, чем технология. В этой книге мы хотим предложить свежий взгляд именно на эту проблему.

Не только для технарей

Было время, когда безопасность беспокоила только государство да движущихся технарей. Но теперь, когда Интернет прочно вошел в жизнь людей по всему миру, обеспечение безопасности лежащих в его основе технологий тревожит часть общества, куда большую, чем когда-либо прежде.

Если вы пользуетесь технологиями, то безопасность имеет для вас значение, потому что любой изъян в ней может нанести прямой ущерб как вам, так и вашему окружению.

Если вы создаете технологию, то должны стремиться сделать ее стабильной и безопасной, чтобы с ее помощью мы могли улучшить бизнес и общество. Безопасность уже не является вопросом, который можно оставить где-то на периферии сознания.

- Вы несете ответственность за безопасность технологии.
- Вы заботитесь о том, чтобы люди прониклись идеей безопасности в повседневной жизни.

Если вы не берете на себя такую ответственность, значит, создаваемая вами технология фундаментально ущербна и не выполняет одну из своих главных функций.

Безопасность и риск неразделимы

Безопасность, а точнее безопасность программного обеспечения, призвана минимизировать риск. Смысл ее в том, что мы пытаемся уменьшить вероятность того, что людей, системы и данные можно будет использовать таким способом, который причинит финансовый или материальный ущерб или нанесет урон репутации организации.

Уязвимость: вероятность и последствия

Большинство методов обеспечения безопасности направлено на предотвращение атак на наши системы и информацию. Но расчет риска – это не попытка предотвратить событие, а стремление понять, что и как

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru