Предисловие

Команда CERT была организована Управлением перспективных исследовательских программ в области обороны (Defense Advanced Research Projects Agency — DARPA) в ноябре 1988 года в ответ на инцидент с червем Морриса, который привел в неработоспособное состояние 10% всех интернет-систем. CERT располагается в Питтсбурге, штат Пенсильвания, в Институте программной инженерии (SEI), и находится под эгидой Министерства обороны США.

Изначально основной задачей CERT было реагирование на инциденты и их анализ. К инцидентам относятся успешные атаки, такие как утечка данных и отказ в обслуживании, а также попытки атак, зондирование и сканирование. С 1988 года CERT получила более 22 665 обращений, связанных с инцидентами в области компьютерной безопасности или с просьбой предоставить информацию, и обработала более 319 992 инцидентов, связанных с безопасностью компьютеров. И это количество с каждым годом растет все быстрее.

Простой реакции на инциденты при их выявлении недостаточно для обеспечения безопасности Интернета и взаимосвязанных информационных систем. Анализ показывает, что большинство инцидентов вызвано троянскими программами, применением методов социальной инженерии и эксплуатацией уязвимостей программного обеспечения, включая дефекты программного обеспечения, проектные решения, конфигурации решений и неожиданные взаимодействия между системами. CERT проводит мониторинг открытых источников по уязвимости информации и регулярно получает сообщения о новых уязвимостях. С 1995 года было зарегистрировано свыше 16 726 уязвимостей. При получении информации CERT анализирует потенциальную уязвимость и сотрудничает с разработчиками технологий, чтобы информировать их о недостатках безопасности в их продукции.[3]

[3] CERT работает более чем с 1900 разработчиками аппаратного и программного обеспечения.

Аналогично сообщениям об инцидентах растет количество сообщений об уязвимостях.[4] Простой работы с обнаруженными уязвимостями недостаточно для решения проблем безопасности Интернета и информационных систем. Становится все более очевидным, что проблема растущего числа уязвимостей и инцидентов должна решаться в их источнике, работать надо в первую очередь над предотвращением появления уязвимостей программного обеспечения во время разработки программного обеспечения и его обслуживания. Анализ существующих уязвимостей показывает, что причин большинства из них относительно немного. *Цель этой книги — ознакомить разработчиков с этими причинами и рассказать о шагах, которые могут быть предприняты с тем, чтобы уязвимости не могли возникнуть.*

Целевая аудитория

Эта книга будет полезна всем, кто участвует в разработке или сопровождении программного обеспечения на языках программирования C и C++.

- Если вы *программист на C/C++*, эта книга научит вас выявлять наиболее распространенные программные ошибки, которые приводят к уязвимости программного обеспечения, понимать, как эти ошибки используются злоумышленниками и как реализовывать свои решения безопасным образом.
- Если вы *руководитель программного проекта*, эта книга поможет вам выявить риски и последствия уязвимости программного обеспечения и укажет основные направления деятельности по разработке безопасного программного обеспечения.
- Если вы *студент-кибернетик*, эта книга научит вас приемам программирования, которые помогут вам избежать развития вредных привычек и позволят разрабатывать безопасные программы во время вашей профессиональной карьеры.
- Если вы *аналитик в области безопасности*, то в этой книге вы найдете подробное описание распространенных уязвимостей; здесь же определяются пути обнаружения уязвимостей и предлагаются практические стратегии предотвращения уязвимостей.

Организация книги

Эта книга — практическое руководство по безопасному программированию на языках программирования C и C++. Получить безопасную программу без безопасного проектирования невозможно.

Однако даже наилучший проект может привести к небезопасной программе, если разработчики не знают о многих ошибках безопасности, присущих программированию на C и C++. В этой книге подробно объясняются распространенные ошибки программирования в C и C++ и описывается, как эти ошибки могут привести к возникновению уязвимого кода. Книга концентрируется на вопросах безопасности для языков программирования C и C++ и связанных с ними библиотек. Она *не* касается вопросов безопасности взаимодействия с внешними системами, например баз данных и веб-серверов, — это достаточно обширные самостоятельные темы. Цель книги — быть полезной для всех, кто участвует в разработке безопасных программ на C и C++ независимо от конкретного приложения.

Книга организована вокруг обычно реализуемых программистами функциональных возможностей, которые имеют потенциальные последствия для безопасности, такие как форматированный вывод и арифметические операции. Каждая глава описывает

[4] Текущая статистика приведена по адресу `www.cert.org/stats/cert_stats.html`.

небезопасные практики программирования и распространенные ошибки, которые могут привести к уязвимости, каким образом эти недостатки программирования могут быть использованы, каковы потенциальные последствия этого использования уязвимости и какие имеются безопасные альтернативы. Здесь рассказывается об основных причинах уязвимости программного обеспечения, таких как переполнение буфера, ошибки диапазона целочисленных типов и недопустимый формат строки. В каждой главе описаны стратегии безопасной реализации функциональных возможностей, а также методы обнаружения уязвимостей в существующем коде.

Книга состоит из следующих глав.

- **Глава 1, "Бег с ножницами"**, представляет собой введение в проблему, вводит основные термины и концепции и рассказывает о том, почему в программах на C и C++ так часто встречаются уязвимости.
- **Глава 2, "Строки"**, описывает работу со строками в C и C++, распространенные недостатки безопасности и получающиеся в результате уязвимости, включая переполнение буфера и нарушение стека.
- **Глава 3, "Уловки с указателями"**, знакомит с *записью в произвольную область памяти*, которая позволяет атакующему записать отдельный адрес в любое место в памяти. В этой главе описывается применение данной уязвимости для выполнения произвольного кода на атакованной машине. Уязвимости в результате записи в произвольное место памяти рассматриваются и в последующих главах.
- **Глава 4, "Управление динамической памятью"**, описывает динамическое управление памятью. Здесь описаны уязвимости, связанные с переполнением динамически выделенного буфера, записью в освобожденную память и двойным освобождением памяти.
- **Глава 5, "Целочисленная безопасность"**, охватывает вопросы целочисленной безопасности (связанной с работами с целыми числами), включая переполнение, ошибки, связанные со знаком числа и с его усечением.
- **Глава 6, "Форматированный вывод"**, описывает корректное и некорректное применение функций форматированного вывода. В главе описаны уязвимости, связанные с форматными строками и переполнением буфера, являющиеся результатом некорректного применения указанных функций.
- **Глава 7, "Параллельное выполнение"**, обращается к вопросам параллельных вычислений и к уязвимостям, причиной которых могут быть взаимоблокировки, гонка данных и некорректные последовательности обращения к памяти.
- **Глава 8, "Файловый ввод-вывод"**, описывает распространенные уязвимости, связанные с файловым вводом-выводом, включая уязвимости, связанные с гонкой данных и временем проверки и использования (time of check, time of use — TOCTOU).
- **Глава 9, "Рекомендованные практики"**, предлагает определенные практики разработки для повышения общего уровня безопасности приложений на языках программирования C/C++. Эти рекомендации дополняют советы, приводимые в каждой главе для уязвимостей определенного класса.

В данной книге описаны сотни примеров безопасного и небезопасного кода, а также образцы использования уязвимостей. Почти все эти примеры — на языках программирования

C и C++, хотя имеются сравнения и с другими языками. Примеры реализованы для операционных систем Windows и Linux. В то время как конкретные примеры обычно компилируются и испытываются в одной или нескольких конкретных средах, для уязвимостей выполняется оценка, позволяющая определить, привязаны ли они к конкретной версии компилятора или не зависят от нее, связаны ли они с конкретной операционной системой, микропроцессором, применяемыми стандартами C или C++, прямым или обратным порядком байтов и архитектурой используемого стека.

Эта книга, как и разработанный на ее базе онлайн-курс, фокусируется на наиболее распространенных программных ошибках при использовании C и C++, которые часто приводят к уязвимости программного обеспечения. Однако из-за ограниченного объема книги рассмотрены далеко не все потенциальные источники уязвимости. Дополнительную и обновленную информацию, расписания событий и новости, связанные с книгой, доступны по адресу `www.cert.org/books/secure-coding/`. Уязвимости, рассматриваемые в книге, связаны перекрестными ссылками с реальными примерами из базы данных US-CERT Vulnerability Notes Database, находящейся по адресу `www.kb.cert.org/vuls/`.

Доступ к онлайн-курсу безопасного кодирования, который связан с этой книгой, предоставляется через Carnegie Mellon's Open Learning Initiative (OLI) по адресу `https://oli.cmu.edu/`. Ключ курса — 0321822137.

Благодарности

Я хотел бы поблагодарить всех, кто сделал возможным выход этой книги в свет. В первую очередь, это Нупур Дэвис (Noopur Davis), Чад Дугерти (Chad Dougherty), Дуг Гвин (Doug Gwyn), Дэвид Китон (David Keaton), Фред Лонг (Fred Long), Нэнси Мид (Nancy Mead), Роберт Мид (Robert Mead), Герхард Мюнц (Gerhard Muenz), Роб Муравски (Rob Murawski), Дэниел Плакош (Daniel Plakosh), Джейсон Рафаил (Jason Rafail), Дэвид Рили (David Riley), Мартин Себор (Martin Sebor) и Дэвид Свобода (David Svoboda), которым я признателен за непосредственный вклад в главы данной книги. Я также хотел бы поблагодарить за их исследования следующих людей: Омар Алхазми (Omar Alhazmi), Арчи Эндрюс (Archie Andrews), Мэтью Коновер (Matthew Conover), Джеффри С. Дженнари (Jeffrey S. Gennari), Одед Горовиц (Oded Horovitz), Пол-Хеннинг Камп (Poul-Henning Kamp), Дуг Ли (Doug Lea), Яшвант Малайя (Yashwant Malaiya), Джон Роберт (John Robert) и Тим Вильсон (Tim Wilson).

Я выражаю благодарность менеджерам SEI и CERT, которые помогали мне в моей работе: Джеффри Карпентеру (Jeffrey Carpenter), Джеффри Хаврилла (Jeffrey Havrilla), Швану Хернану (Shawn Hernan), Ричу Пети (Rich Pethia) и Биллу Вильсону (Bill Wilson).

Большое спасибо моему редактору Питеру Гордону (Peter Gordon) и команде из Addison-Wesley: Дженнифер Эндрюс (Jennifer Andrews), Ким Бодигхаймер (Kim Boedigheimer), Джон Фуллер (John Fuller), Эрик Гарулей (Eric Garulay), Стефан Накиб (Stephane Nakib), Элизабет Райан (Elizabeth Ryan) и Барбара Вуд (Barbara Wood).

Я признателен всем, кто помогал в разработке курса Open Learning Initiative, и в первую очередь — Марше Ловетт (Marsha Lovett), Норману Биру (Norman Bier) и Александре Дрозд (Alexandra Drozd).

Хочу также поблагодарить следующих рецензентов за комментарии и замечания: Тэд Андерсон (Tad Anderson), Джон Бенито (John Benito), Уильям Балли (William Bulley), Кори Коген (Corey Cohen), Уилл Дорманн (Will Dormann), Уильям Фитен (William Fithen), Робин Эрик Фредериксен (Robin Eric Fredericksen), Майкл Говард (Michael Howard), Майкл Кейлблинг (Michael Kaelbling), Амит Калани (Amit Kalani), Джон Ламберт (John Lambert),

Джеффри Ланца (Jeffrey Lanza), Дэвид ЛеБланк (David LeBlanc), Кен Мак-Иннис (Ken MacInnis), Гэри Мак-Гроу (Gary McGraw), Рэнди Мейерс (Randy Meyers), Филип Миллер (Philip Miller), Патрик Мюллер (Patrick Mueller), Дэйв Мунди (Dave Mundie), Крейг Партридж (Craig Partridge), Брэд Руббо (Brad Rubbo), Тим Шимилл (Tim Shimeall), Майкл Ванг (Michael Wang) и Кэти Вашок (Katie Washok).

Хочу поблагодарить за помощь и поддержку всех неупомянутых членов команды CERT, без которых я никогда не смог бы завершить работу над этой книгой. И — последними по списку, но не по важности — я хотел бы выразить благодарность нашим редакторам и библиотекарям, помогавшим мне в работе. Это Рэчел Каллисон (Rachel Callison), Памела Кертис (Pamela Curtis), Лен Эстрин (Len Estrin), Эрик Хайес (Eric Hayes), Кэрол Дж. Лалье (Carol J. Lallier), Карен Рили (Karen Riley), Шейла Розенталь (Sheila Rosenthal), Пенни Уолтерс (Pennie Walters) и Барбара Уайт (Barbara White).