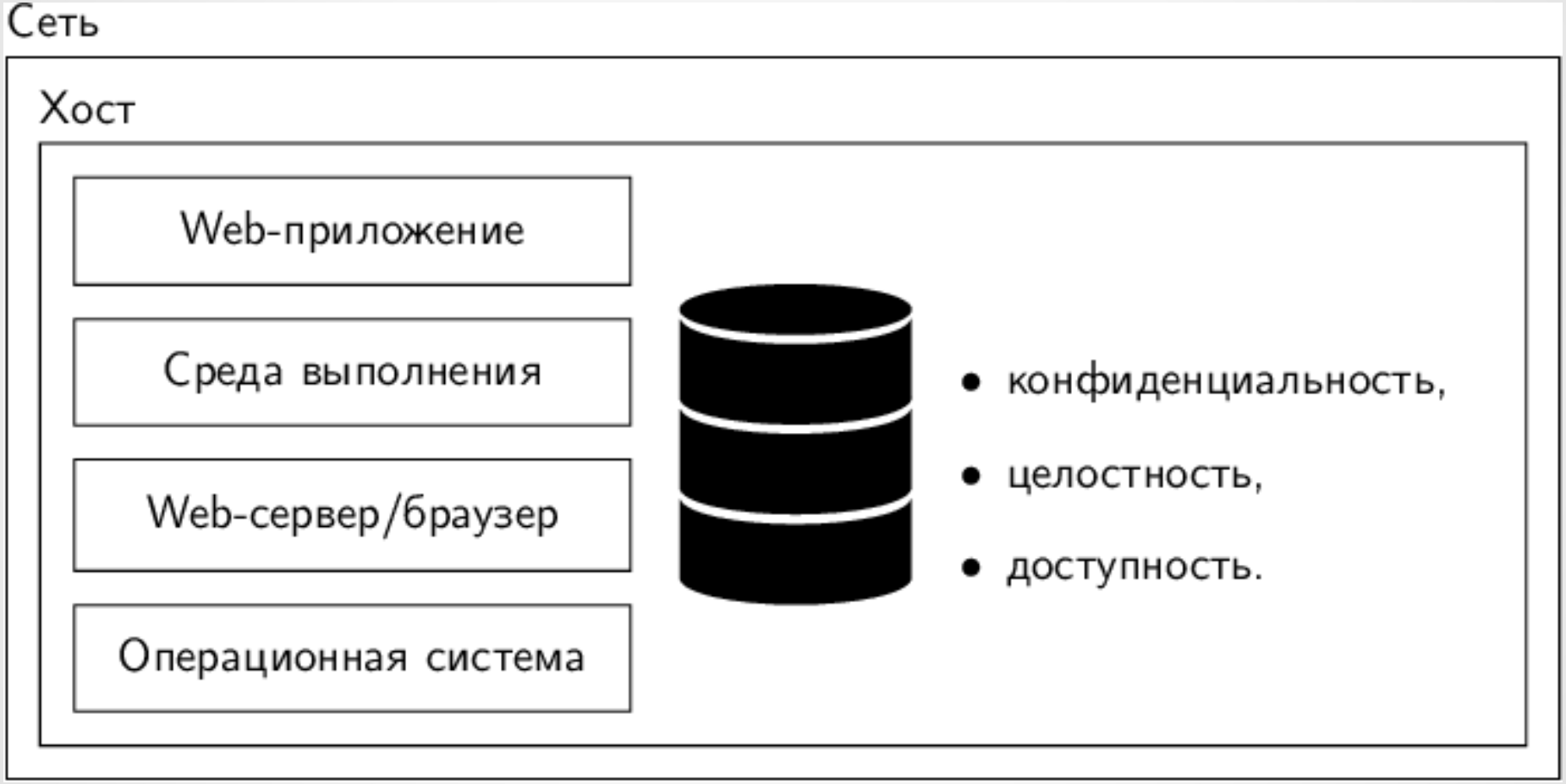
Основы информационной безопасности веб-приложений

Кулаков Кирилл Александрович

Базовые понятия



Примеры уязвимостей

- https://owasp.org/www-community/vulnerabilities/
- Отсутствие/недостаточная валидация пользовательского ввода
- Валидация пользовательского ввода на стороне клиента
- Некорректная обработка ошибок
- Утечка информации о внутреннем устройстве приложения
- Хранение паролей в открытом виде
- Недостаточно длинные/случайные идентификаторы сессий

Примеры уязвимостей

- Отсутствие/недостаточная валидация пользовательского ввода
  - Формы с одинаковым названием
  - ошибочный метод validate ()
  - форма не расширяет класс проверки
  - поле формы без валидатора
  - неиспользованная форма проверки
  - форма непроверенного действия
  - валидатор выключен
  - валидатор без поля формы

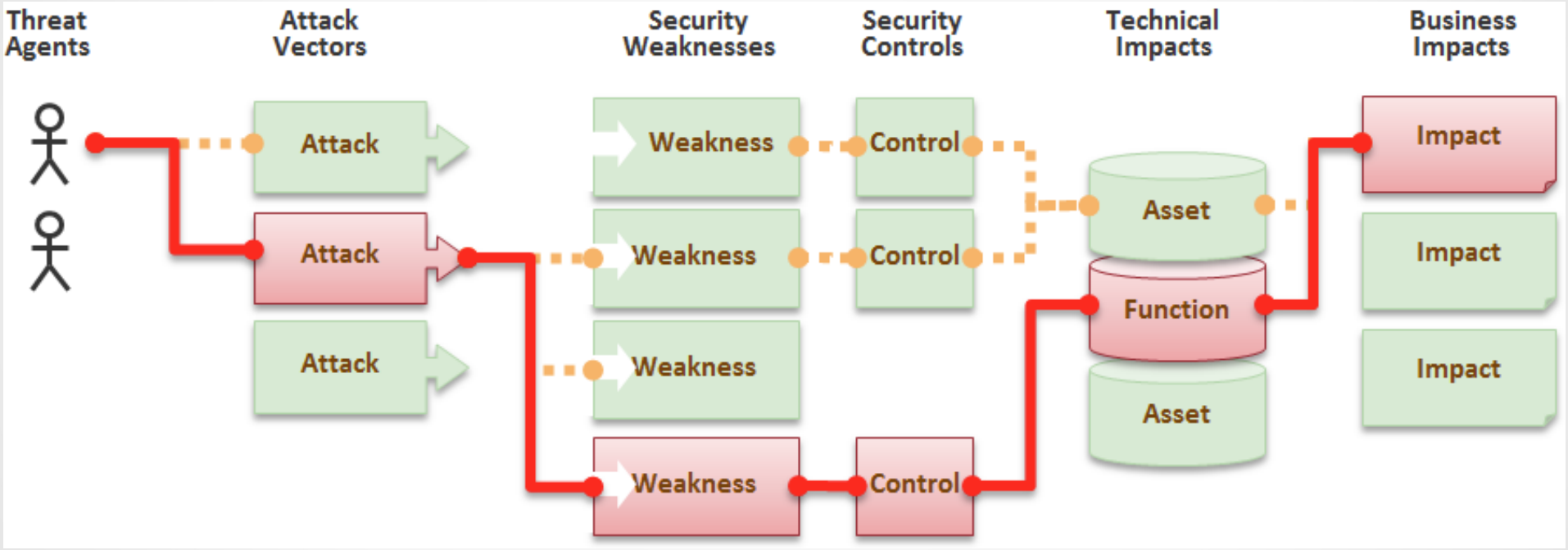
Примеры атак

- https://owasp.org/www-community/attacks/
- SQL-injection (проверка данных в запросах к БД)
- Path traversal attack (проверка пути)
- Cross-Site Scripting (XSS) (кто-то может передать браузеру злонамеренный код от имени приложения)
- Phishing (контроль запросов со сторонних сайтов)
- Cross-Site Request Forgery (CSRF) (кто-то может отправить запрос от имени пользователя)
- Denial of Service (DOS) (следите за выделением/освобождением ресурсов)
- Man in the Middle (MITM) (перехват сообщений от клиента к серверу, HTTPS, HeartBleed (доступ через SSL) и прочие радости)

Контрмеры

- Authentication (Идентификация)
- Authorization / Access Control (Проверка доступа)
- Audit (Аудит безопасности)
- Session Management (Управление сессиями)
- Input Validation (Проверка ввода)
- Error Handling (Обработка ошибок)
- Logging (Логирование операций)
- Cryptography (Криптография)
- Application Software Security (Контроль разработки и обеспечение качества ПО)

Анализ рисков информационной безопасности



Топ 10 проблем безопасности

- https://owasp.org/www-project-top-ten/ (2017)
- **Инъекция**. Недостатки внедрения, такие как внедрение SQL, NoSQL, ОС и LDAP, возникают, когда ненадежные данные отправляются интерпретатору как часть команды или запроса. Враждебные данные злоумышленника могут обманом заставить интерпретатор выполнить непредусмотренные команды или получить доступ к данным без надлежащей авторизации.
- **Сломанная аутентификация**. Функции приложений, связанные с аутентификацией и управлением сеансами, часто реализуются неправильно, что позволяет злоумышленникам скомпрометировать пароли, ключи или токены сеанса или использовать другие недостатки реализации для временного или постоянного присвоения идентификаторов других пользователей.

Топ 10 проблем безопасности

- **Раскрытие конфиденциальных данных**. Многие веб-приложения и API не защищают должным образом конфиденциальные данные, такие как финансовые, медицинские и PII. Злоумышленники могут украсть или изменить такие слабо защищенные данные для совершения мошенничества с кредитными картами, кражи личных данных или других преступлений. Конфиденциальные данные могут быть скомпрометированы без дополнительной защиты, такой как шифрование при хранении или передаче, и требуют особых мер предосторожности при обмене с браузером.
- **Внешние объекты XML (XXE)**. Многие старые или плохо настроенные процессоры XML оценивают ссылки на внешние объекты в документах XML. Внешние объекты могут использоваться для раскрытия внутренних файлов с помощью обработчика URI файла, внутренних общих файловых ресурсов, сканирования внутреннего порта, удаленного выполнения кода и атак типа «отказ в обслуживании».

Топ 10 проблем безопасности

- **Нарушенный контроль доступа**. Ограничения на то, что разрешено делать аутентифицированным пользователям, часто не соблюдаются должным образом. Злоумышленники могут использовать эти недостатки для доступа к неавторизованным функциям и / или данным, например для доступа к учетным записям других пользователей, просмотра конфиденциальных файлов, изменения данных других пользователей, изменения прав доступа и т. Д.
- **Неверная конфигурация безопасности**. Неверная конфигурация безопасности - наиболее часто встречающаяся проблема. Обычно это является результатом небезопасных конфигураций по умолчанию, неполных или специальных конфигураций, открытого облачного хранилища, неправильно настроенных заголовков HTTP и подробных сообщений об ошибках, содержащих конфиденциальную информацию. Все операционные системы, платформы, библиотеки и приложения должны быть не только надежно настроены, но и своевременно исправлены / обновлены.

Топ 10 проблем безопасности

- **Межсайтовый скриптинг XSS**. Недостатки XSS возникают всякий раз, когда приложение включает ненадежные данные на новую веб-страницу без надлежащей проверки или экранирования или обновляет существующую веб-страницу с данными, предоставленными пользователем, с помощью API браузера, который может создавать HTML или JavaScript. XSS позволяет злоумышленникам выполнять сценарии в браузере жертвы, которые могут перехватывать сеансы пользователя, портить веб-сайты или перенаправлять пользователя на вредоносные сайты.
- **Небезопасная десериализация**. Небезопасная десериализация часто приводит к удаленному выполнению кода. Даже если недостатки десериализации не приводят к удаленному выполнению кода, их можно использовать для выполнения атак, включая атаки с повторением, атаки с использованием инъекций и атаки с повышением привилегий.

Топ 10 проблем безопасности

- **Использование компонентов с известными уязвимостями**. Компоненты, такие как библиотеки, фреймворки и другие программные модули, работают с теми же привилегиями, что и приложение. В случае использования уязвимого компонента такая атака может способствовать серьезной потере данных или захвату сервера. Приложения и API-интерфейсы, использующие компоненты с известными уязвимостями, могут подорвать защиту приложений и сделать возможными различные атаки и воздействия.
- **Недостаточное ведение журнала и мониторинг**. Недостаточное ведение журнала и мониторинг в сочетании с отсутствующей или неэффективной интеграцией с реагированием на инциденты позволяет злоумышленникам продолжать атаковать системы, поддерживать постоянство, переходить к большему количеству систем и вмешиваться, извлекать или уничтожать данные. Большинство исследований показывают, что время обнаружения нарушения составляет более 200 дней, и обычно оно обнаруживается внешними сторонами, а не внутренними процессами или мониторингом.

Рекомендуемые практики

- Apply defense in depth (Глубокая защита)
- Use a positive security model (Доступно что разрешено)
- Fail securely (Безопасная обработка ошибок)
- Run with least privilege (Минимум привелегий)
- Avoid security by obscurity (безопасность не зависит от секретов)
- Keep security simple (Простота безопасности)
- Detect intrusions (Обнаружение вторжений)
- Don’t trust infrastructure (Нет доверия окружению)
- Don’t trust services (Нет доверия сервисам)
- Establish secure defaults (безопасные настройки по умолчанию)

Практики контроля сессий

- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
- Идентификатор сеанса не должен содержать описание
- Длина идентификатора сеанса должна быть не менее 128 бит
- Идентификатор сеанса должен быть непредсказуемым (достаточно случайным)
- Содержимое (или значение) идентификатора сеанса не должно иметь смысла для предотвращения атак раскрытия информации
- Управление сеансом по правилам фреймворка
- Использование cookie для обмена идентификаторами
- Использование шифрованных соединений
- Идентификаторы сеанса должны считаться ненадежными, как и любой другой пользовательский ввод
- Идентификатор сеанса должен быть обновлен или восстановлен веб-приложением после любого изменения уровня привилегий
- Автоматическое истечение срока сеанса
- Обнаружение атак

Практики работы с паролями

- https://crackstation.net/hashing-security.htm
- Хеширование паролей
  - Словарь и атаки грубой силы
  - Таблицы поиска
  - Таблицы обратного просмотра
  - Радужные столы
- Добавление соли
  - Повторное использование соли
  - Короткая соль
- Хеш-коллизии
  - перемешивание с солью
- медленные хеш-функции
- хэши с ключами и оборудование для хеширования паролей