Примерные вопросы к экзамену
по МДК.01.01 «Эксплуатация подсистем безопасности автоматизированных систем»
для студентов групп БИ-1-15 и БИ-2-15
(1-й семестр 2017/2018 учебного года)

Преподаватель, составитель: Караваев Сергей Владимирович. В каждом экзаменационном билете — два теоретических вопроса и одно практическое задание.

Примерные теоретические вопросы

1. Безопасность автоматизированной информационной системы: определение по Р 50.1.053—2005, конфиденциальность, доступность, целостность, подотчётность и подлинность ресурсов.
2. Нормативно-правовые акты Российской Федерации в сфере информационной безопасности.
3. Электронный документооборот: определение, классификация систем автоматизации документооборота, угрозы безопасности.
4. Модель угроз безопасности и нарушителя безопасности автоматизированной системы.
5. Простая аутентификация субъекта доступа по ГОСТ Р ИСО/МЭК 9594-8—98: определение, принципы, уязвимости. Привести описание подсистемы аутентификации, применяемой в Laravel.
6. Авторизация в автоматизированной системе. Рассмотреть дискреционное (избирательное) управление доступом (DAC) на примере подсистемы политик (англ. policies) в Laravel.
7. Атака по обходу каталогов (англ. path traversal attack): векторы атаки, методика противодействия. SEO-friendly URL: определение, способы реализации.
8. Статусы и заголовки ответа HTTP/1.1. Привести примеры управления заголовками HTTP/1.1.
9. Hardening сервера HTTP и транслятора серверного языка программирования.
10. HTTP over TLS (HTTP/TLS): назначение, задействованные уровни модели OSI, схема https.
11. Secure cookie protocol: определение, услуги. Привести пример реализации.
12. Методы HTTP GET и POST: назначение, ограничения, идемпотентность. Сравнить типы содержимого `application/x-www-form-urlencoded`, `multipart/form-data`, `text/plain`. Привести пример сообщения.
13. Контроль ввода информации в АС: средства проверки в полях ввода в HTML 5.1.
14. Контроль ввода информации в АС: подсистема Request в Laravel.
15. Аудит веб-ресурса на соответствие рекомендациям W3C.
16. Шаблонизация в веб-разработке. Принцип работы шаблонизатора. Назвать один из наиболее популярных шаблонизаторов, привести пример его использования.
17. Безопасность паролей: оценка сложности, защита от атак на криптографические хеш-функции (в т. ч. по радужным таблицам), порядок сброса.
18. Исключительная ситуация: определение по ГОСТ 28397—89, порождение, обработка, регенерация.
19. Регистрация аварийных ситуаций в автоматизированной системе.
20. SQL-инъекции Ⅰ порядка: векторы атаки через веб-приложение, методика противодействия, примеры.
21. SQL-инъекции Ⅱ порядка: векторы атаки через веб-приложение, методика противодействия, примеры.

22. Отражённые (англ. reflected) и хранимые (англ. stored) XSS: векторы атаки, методика противодействия. Привести примеры.

23. Подделка межсайтового запроса (CSRF): векторы атаки, методика противодействия, примеры.

24. Man-in-the-middle (MITM): векторы атаки, методика противодействия. Привести примеры.

25. Эксплуатация средств автоматизированного тестирования на проникновение.

Примерные практические задания

1. Продемонстрируйте работу подсистемы политик в Laravel в отношении моделей «Блогер» («Пользователь») и «Запись в блоге». Блогер имеет право прочитать любую запись в блоге, но редактировать и удалять может только собственные записи.

2. Продемонстрируйте работу подсистемы политик в Laravel в отношении моделей «Пользователь» и «Личные сообщения». Право читать личное сообщение имеют отправитель и адресат, редактировать и удалять личное сообщение запрещено.

3. Реализуйте и введите в эксплуатацию модуль, например класс-наследник Request в Laravel, обеспечивающей контроль ввода информации в модель «Товар». Задать следующие ограничения: значение атрибута «Наименование» — строковое, обязательное, уникальное, длиной 1–255 знаков; значение атрибута «Цена» — числовое неотрицательное. Все атрибуты обязательные.

4. Реализуйте и введите в эксплуатацию модуль, например класс-наследник Request в Laravel, обеспечивающей контроль ввода информации в модель «Учётные записи физических лиц в налоговом органе». Задать следующие ограничения: значение атрибута «ИНН» — обязательное, состоит из 12 арабских цифр; значение атрибута «Дата выдачи» — любого формата даты. Все атрибуты обязательные.

5. Реализуйте и введите в эксплуатацию модуль, например класс-наследник Request в Laravel, обеспечивающей контроль ввода информации в модель «Отделения почты». Задать следующие ограничения: значение атрибута «Почтовый индекс» — обязательное, состоит из 6 арабских цифр; значение атрибута «Адрес» — строковое, длиной 1–255 знаков. Все атрибуты обязательные.

6. Реализуйте и введите в эксплуатацию шаблон (совместимый, например, с шаблонизатором Blade) для вывода перечня записей в блоге/пользователей/товаров в формате HTML. Обеспечьте кодирование данных для защиты от хранимых XSS.

7. Выполняется запрос: `SELECT * FROM users WHERE username = '$username' AND password = '$password';` (значения `$username` и `$password` интерполируются без фильтрации). Проведите SQL-инъекцию I порядка, которая приведёт к выборке всех кортежей. Приведите фрагмент исходного кода приложения базы данных, который предотвращает SQL-инъекции I порядка.

8. Реализуйте и введите в эксплуатацию шаблон формы HTML (совместимый, например, с шаблонизатором Blade) для редактирования кортежей в таблице «Товары» реляционной базы данных. Обеспечьте защиту от CSRF с помощью токенов и от SQL-инъекции I порядка — с помощью подготовленных предложений SQL.

9. Реализуйте и введите в эксплуатацию шаблон формы HTML (совместимый, например, с шаблонизатором Blade) для редактирования кортежей в таблице «Записи в блоге» реляционной базы данных. Обеспечьте защиту от CSRF с помощью токенов и от SQL-инъекции I порядка — с помощью подготовленных предложений SQL.

10. Создать самозаверенный сертификат и продемонстрируйте с применением Wireshark работу протокола HTTP over TLS на произвольном веб-приложении.