Фонд оценочных средств
для промежуточной аттестации
по дисциплине
Исследование вредоносного программного обеспечения

Направление подготовки (специальность)
10.05.01 Компьютерная безопасность

Направленность (профиль)
специализация № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Присваиваемая квалификация
специалист по защите информации

Форма обучения
очная

Челябинск 2025 г.

Содержание

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Специальность 10.05.01 Компьютерная безопасность.
Специализация № 6 «Информационно-аналитическая и техническая экспертиза компьютерных систем».
Дисциплина: **Исследование вредоносного программного обеспечения**.
Семестр (семестры) изучения: 10 семестр.
Форма (формы) промежуточной аттестации: зачёт 10 семестр.
Используется балльно-рейтинговая система для оценивания результатов.

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Исследование вредоносного программного обеспечения» направлено на формирование следующих компетенций:

Коды компетенции согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ПК-3	Способен проводить анализ безопасности компьютерных систем	ПК-3.1 Обладает знаниями о уровнях защищенности и доверия в компьютерных системах; об оценках рисков, связанных с осуществлением угроз безопасности в отношении компьютерных систем; об оценках соответствия механизмов безопасности компьютерной системы требованиям существующих нормативных документов, а также их адекватности существующим рискам. ПК-3.2 Демонстрирует умения: проводить мониторинг, анализ и сравнение эффективности программно-аппаратных средств защиты информации в операционных системах; формулировать и разрабатывать предложения по устранению выявленных уязвимостей. ПК-3.3 Имеет практический опыт (навыки): выполнение анализа уязвимости компьютерных систем.	Знать: – методы проникновения в компьютерные системы, используемые современным вредоносным программным обеспечением; – методы функционирования современного вредоносного программного обеспечения. Уметь: – реализовывать современные атаки на компьютерные системы; – исследовать вредоносное программное обеспечение. Владеть: – инструментами проведения современных атак на компьютерные системы; – навыками использования инструментальных средств исследования вредоносного программного обеспечения.

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Код компетенции / планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1.	ПК-3	Раздел 1. Инструменты исследования кода	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте.
2.	ПК-3	Раздел 2. Вредоносное программное обеспечение	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте.
3.	ПК-3	Раздел 3. Безопасность аппаратного обеспечения	Зачетные задания. Домашние задания. Аудиторные задания.	Вопросы на зачёте. Задания на зачёте.

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре

3.2 Содержание оценочных средств

3.2.1 Темы для домашних, аудиторных, зачетных заданий и заданий на экзамене

1. Шеллкоды.
2. Эксплуатация уязвимостей.
3. Методы функционирования вредоносного ПО.
4. Методы исследования вредоносного ПО.
5. Анализ безопасности аппаратного обеспечения.
6. Использование аппаратного обеспечения при проведении атак на компьютерные системы.

3.2.2 Примеры зачетных заданий.

1. Реализовать плагин для IDA.

Плагин должен предоставлять следующие возможности по динамическому анализу кода:

1) трассировка кода на уровне инструкций;
2) трассировка кода на уровне функций (с выводом аргументов);
3) инструментация кода на уровне функций;
4) перехват библиотечных вызовов;
5) возможность модификации поведения и аргументов перехваченных функций;
6) отслеживание работы с памятью;
7) отслеживание модификаций кода.

25 баллов.

2. Реализовать атаку BadUSB на плате с чипом ATmega32u4.

В зависимости от положения переключателей на плате предусмотреть следующие действия:

- ничего не делать;
- запускать на исполнение программу;
- открывать в браузере сайт;
- скачивать по сети программу и запускать;
- другое.

25 баллов.

3. Модификация прошивки роутера.

Внести изменения в исходный код и собрать прошивку с бекдором,

предоставляющим следующий функционал:

1) предоставление удалённой оболочки по сети;
2) возможность передачи файлов;
3) возможность обновления прошивки;
4) защита от перезаписи прошивки.

25 баллов.

4. Буткит.

Реализовать буткит для процессоров архитектуры IA-32. Расположить буткит в ROM BIOS сетевой карты. Буткит должен предоставлять следующий функционал:

1) трассировка кода загрузчика;
2) перехват функций работы с диском;
3) перехват загрузки ядра;
4) внесение модификаций в ядро при чтении с диска.

25 баллов.

3.2.3 Примеры домашних, аудиторных заданий и заданий на экзамене

1. Написать плагин для IDA, сохраняющий дизассемблерный листинг функций в файл.

2. Написать скрипт, расшифровывающий код в примере test.c.

3. Дизассемблировать task\task1.exe и добиться корректной аутентификации.

4. Плагин, подсчитывающий побайтную энтропию различных участков исполняемого файла. По возможности предложить алгоритм, определяющий участки с одинаковой энтропией.

5. Плагин, определяющий для каждой функции доступные из неё импортируемые функции через цепочку вызова. Импортируемая функция imp_fun, считается доступной из некоторой функции fun, если fun вызывает imp_fun непосредственно или вызывает другую функцию other_fun, из которой доступна imp_fun.

Последовательность действий может быть следующая. Найти все импортируемые имена. Найти все перекрестные ссылки на эти имена. Это будут функции, из которых импортируемые имена доступны непосредственно. Запомнить эту информацию. Затем для этих функций рассмотреть все перекрестные ссылки. Это будут функции, из которых импортируемые функции доступны через один вызов. Запомнить информацию для них. И т.д. Т.е. реализовать поиск в ширину в обратном

графе вызовов от множества импортируемых функций. Когда поиск будет закончен сохранить информацию о доступных импортируемых функциях в комментариях для каждой функции.

6. Написать скрипт на Python для IDA.

Пробежать по всем функциям. Вывести для каждой функции список перекрестных ссылок на её начало. Затем пробежать по каждой инструкции функции и вывести список перекрестных ссылок на инструкцию.

7. Написать скрипт на Python для IDA.

Для любой выбранной функции находить все функции из неё достижимые.

8. На основе плагина idasploiter.py реализовать плагин поиска гаджетов в исполняемой секции загруженной программы без запуска этой программы.

9. Реализовать плагин для IDA.

Установить точки на останова на импортируемые функции. При срабатывании точки останова выводить аргументы функций (некоторых известных, например, CreateFile, CreateProcess) в соответствии с их типом (например, для char* выводить строку).

10. Реализовать плагин для IDA.

Трассировать программу (запрашивать у пользователя начальный и конечный адрес) и сохранять в файл каждую инструкцию и значение регистров на каждом шаге.

11. Реализовать плагин для IDA.

Реализовать трассировку (самостоятельно ставить точки останова) программы на уровне функций. Необходимо отслеживать все функции программы и импортируемые из неё. Для каждого вызова функции выводить информацию об аргументах (когда возможно, разыменовывать указатели).

12. Плагин для отладчика, подменяющий прочитанные из некоторого файла данные.

Для этого может потребоваться перехватывать функции CreateFile, ReadFile, CloseHandle, MapViewOfFile и получать управление после возврата из них (для сохранения возвращаемых данных или смены данных в памяти).

13. Реализовать плагин для IDA.

Реализовать трассировку всех инструкций. Во время трассировки отлавливать и логировать в файл все инструкции доступа (чтения, записи) к глобальной памяти программы.

14. Модифицировать пример SpectrePoC-my_modification.

1) Внести в кэш размер массива.

2) Изменить размер блока в кэшируемом массиве.

3) Изменить количество тренировочных вызовов.

4) Изменить задержки после сброса кэша.

5) Убрать перемешивание при проверке времени доступа к кэшированным данным.

6) Изменить (уменьшить) количество попыток закэшировать считываемый байт (сейчас 5 попыток, на каждой 5 тренировочных и одна атакующая).

7) Добавить код (разные инструкции) после проверки условия (внутри блока исполнения if) в функции VictimFunction. Добиться того, чтобы спекулятивное исполнение не доходило до утечки памяти.

14. Перехватывать прерывание BIOS чтения с диск (int 13h). При попытке чтения выводить отладочную информацию (адрес на диске, адрес буфера, количество секторов).

15. Написать отладочный плагин для IDA.

С помощью него отладить пример loader_move_int13. Поставить точку останова на наш обработчик int 13h (перед вызовом настоящего обработчика). Выводить аргументы чтения с диска. Определить момент считывания ядра.

16. Реализовать трассировку кода загрузчика (несколько первых инструкций).

17. Реализовать проекты для Arduino (конкретная функциональность определяется самостоятельно).

1) Подключить светодиод (к цифровому выводу) и управлять им.

2) Обрабатывать нажатия кнопок.

3) Организовать связь двух плат (через цифровые выводы).

4) Управлять светодиодами с помощью микросхемы SN74HC595N.

5) Реализовать интерактивное взаимодействие с помощью кнопок и монитора.

6) Загрузить в одну плату пример ArduinoISP и с помощью неё прошить другую плату.

18. Модифицировать исходный код ядра (ap123/linux/kernels/mips-linux-2.6.31).

Модифицировать системный вызов open так, чтобы при открытии файлов отображались их имена.

19. Написать шеллкод на MIPS, предоставляющий удалённую

оболочку.

20. Реализовать взаимодействие Arduino UNO и ESP-01.

3.2.4 Примеры вопросов на зачете

1. Руткиты.
2. Буткиты.
3. Использованием аппаратной виртуализации вредоносным программным обеспечением.
4. Методы исследования вредоносного программного обеспечения.
5. Инструменты исследования вредоносного программного обеспечения.
6. Скрипты и плагины для дизассемблеров.
7. Скрипты и плагины для отладчиков.
8. Скрипты для Radare2.
9. Анализ безопасности аппаратного обеспечения.
10. Использование аппаратного обеспечения при проведении атак на компьютерные системы.

3.2.5 Пример билета

1. Написать отладочный плагин для IDA.
2. Методы исследования вредоносного программного обеспечения.

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1. Порядок проведения промежуточной аттестации

За своевременное и самостоятельное выполнение учебных работ в течение семестра студент получает рейтинговые баллы. Сумма за выполнение основных заданий в полном объёме – 100. Сверх этой суммы могут начисляться баллы за выполнение дополнительных заданий.

Пропуск по неуважительной причине одной пары влечет вычет 1 балла из итоговой суммы за семестр.

При нехватке баллов преподавателем может быть предоставлено дополнительное задание или возможность доделать задание, в котором была оценена не вся функциональность.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств.

4.2.1 Критерии оценивания зачётных заданий

Основные баллы выставляются за выполнение объемных зачётных заданий, которые выполняются дома и сдаются в течение семестра. Сумма в 100 баллов делится между зачётными заданиями (не обязательно равномерно). При выдаче зачётного задания определено сколько баллов выставляется за реализацию определённой функциональности. Для зачётных заданий может быть определена функциональность повышенной сложности, за выполнение которой выставляются дополнительные баллы. Также дополнительные баллы могут быть выставлены по усмотрению преподавателя за особо примечательную реализацию.

При сдаче зачётного задания производится опрос по техническим деталям реализации и по теории, используемой при выполнении заданий. Неудовлетворительный ответ будет означать несамостоятельность выполнения задания, что влечёт выставление 0 баллов за соответствующую функциональность. Если для одного задания это повторяется более 2 раз, то за всё задание выставляется 0 баллов без возможности повторной сдачи.

Выполнение заданий предполагает некоторую программную реализацию, к которой будут предъявляться обычные требования по качеству кода. Код должен быть удобочитаемым, хорошо структурированным, написанным в едином стиле. Иначе возможна сбавка до 5 баллов. За программные ошибки, приводящие к работоспособности кода не для всех возможных случаев, возможна сбавка до 10 баллов (в зависимости от критичности ошибки). За программные ошибки, приводящие к аварийному некорректному завершению программы, возможна сбавка до 10 баллов (в зависимости от критичности ошибки).

По пройденному материалу выдаются небольшие задания для выполнения дома и/или во время семинарских занятий. За эти задания выставляются небольшие дополнительные баллы. Сдавать их можно либо в день выдачи либо на следующем занятии. Домашние и аудиторные задания – это небольшие задания, за которые обычно выставляется 1-2 балла. Они оцениваются атомарно: либо задание выполнено (выставляется указанное при выдаче задания количество баллов), либо не выполнено (0 баллов).

4.2.2 Критерии оценивания зачёта

Ответ на зачёте оценивается по трём параметрам.

1. Построение ответа (структура ответа, грамотность речи, последовательность и т.д.).

Студент самостоятельно правильно выстраивает структуру ответа, изложение последовательное, речь грамотная без оговорок. 2 балла

Изложение студента непоследовательное и обрывочное, взаимосвязи частей ответа не всегда прослеживаются. Раскрытие сути ответа невозможно без уточняющих вопросов. 1 балл.

Студент испытывает существенные трудности при самостоятельном построении ответа, способен только давать краткие ответы на конкретные вопросы. 0 баллов.

2. Фактическая полнота ответа.

Студент правильно ответил на теоретический вопрос билета. Показал отличные знания в рамках усвоенного учебного материала. Ответил на все дополнительные вопросы. 4 балла.

Студент ответил на теоретический вопрос билета с небольшими неточностями. Показал хорошие знания в рамках усвоенного учебного материала. Ответил на большинство дополнительных вопросов. 3 балла.

Студент ответил на теоретический вопрос билета с существенными неточностями. Показал удовлетворительные знания в рамках усвоенного учебного материала. При ответах на дополнительные вопросы было допущено много неточностей. 2 балла.

При ответе на теоретический вопрос билета студент продемонстрировал недостаточный уровень знаний. При ответах на дополнительные вопросы было допущено множество неправильных ответов. 1 балл.

Студент не ответил на вопрос. 0 баллов.

3. Собственный анализ излагаемого материала его оценка в контексте взаимодействия с другими областями, умение применять на практике.

Студент ясно осознаёт место излагаемого материала в общей структуре профессионального знания, знает стандартные примеры использования и предлагает свои, даёт собственные компетентные оценки. 4 балла.

Студент осознаёт взаимосвязи и знает стандартные примеры использования. Допускает неточности при самостоятельном анализе. 3 балла.

Студент в общих чертах осознаёт взаимосвязи и знает стандартные примеры использования. При проведении самостоятельного анализа нуждается в уточняющих вопросах, при этом допускает существенные неточности. 2 балла.

Студент знает основные стандартные примеры использования. Не

осознаёт взаимосвязей с другими областями. 1 балл.

Студент не осознаёт взаимосвязей и практическое приложение излагаемого материала. 0 баллов.

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Итоговая оценка за дисциплину выставляется по результатам выполнения заданий текущего контроля. При необходимости во время зачёта может быть предоставлена возможность получить дополнительные баллы (не более 20), выполнив дополнительные задания и ответив (в устной форме) на вопросы.

Дополнительные задания, выдаваемые на зачёте, являются относительно объемными, за них выставляется до 10-15 баллов. Поэтому к ним применимы описанные выше критерии оценивания зачётных заданий с соответствующей корректировкой баллов: сбавка за некорректную работу до 5 баллов, за аварийное завершение – до 5 баллов.

Перевод рейтинговых баллов в оценки за зачет:
0-60 баллов – не зачтено;
61-100 баллов – зачтено.

Особенности проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяются следующим образом:

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	более 90 баллов	76-90 баллов	61-75 баллов	0-60 баллов
Уровень освоения проверяемых компетенций	высокий	средний	базовый	недостаточный

Специальность 10.05.01 Компьютерная безопасность
Специализация №6 «Информационно-аналитическая и техническая экспертиза компьютерных систем»
Фонд оценочных средств по дисциплине «Исследование вредоносного программного обеспечения»
2025 год набора, очная форма обучения

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе	утверждено 24.02.25	А.А. Саламатов

Ученым советом математического факультета

Протокол заседания № 6 от 20.02.2025

Председатель Ученого совета математического факультета	согласовано	Е.А. Сбродова

Заседанием кафедры компьютерной безопасности и прикладной алгебры

Протокол заседания № 8 от 01.02.2025

Заведующий кафедрой	согласовано	А. Н. Ручай
Автор (составитель)		И. А. Маткин

Структура рабочей программы соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от «13» апреля 2021 г. № 247-1