МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра компьютерной безопасности и математических методов обработки информации

УТВЕРЖДАЮ

Декан математического факультета

_____ Нестеров П.Н.

20 июня 2023 г.

Рабочая программа дисциплины

Защита программ и данных

Направление подготовки (специальности)
10.04.01 Информационная безопасность

Направленность (профиль)
«Управление информационной безопасностью»

Форма обучения очная

Программа рассмотрена
на заседании кафедры
от 14 апреля 2023 г., протокол № 8

Программа одобрена НМК
математического факультета
протокол № 9 от 3 мая 2023 г.

Ярославль

1. Цель освоения дисциплины.

Целью изучения дисциплины «Защита программ и данных» является теоретическая и практическая подготовка специалистов к деятельности, связанной с применением современных технологий анализа программных реализаций, защиты программ и программных систем от анализа и вредоносных программных воздействий.

2. Место дисциплины в структуре образовательной программы.

Дисциплина «Защита программ и данных» относится к числу дисциплин базовой части профессионального цикла. Для лучшего усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе обучения в бакалавриате по направлениям 10.03.01, 11.03.02, 03.03.03, 11.03.01.

Знания и практические навыки, полученные в результате изучения дисциплины, используются в дисциплинах «Защищенные информационные системы», «Аудит информационной безопасности», «Разработка безопасного программного обеспечения», «Защита от вредоносного программного обеспечения», при разработке курсовых и дипломных работ, и непосредственно в профессиональной деятельности.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы.

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО года для специальности ИБ, направленных на приобретение следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	**Индикатор достижения компетенции (код и формулировка)**	**Перечень планируемых результатов обучения**
Универсальные компетенции		
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий.	**И-УК-1_4** Способен проводить анализ программ на наличие в них вредоносных закладок.	**Уметь:** Проводить анализ программ на наличие в них вредоносных закладок.
	И-УК-1_5 Способен применять патчи безопасности на программы и работающие процессы.	**Уметь:** Применять патчи безопасности на программы и работающие процессы.
Профессиональные компетенции		

ПК-1 Способен разрабатывать математические модели систем обеспечения информационной безопасности, математически доказывать их соответствие выбранным политикам безопасности.	**И-ПК-1_4** Способен для конкретной системы осуществлять обоснованный выбор программных и программно-аппаратных средств, обеспечивающих высокую защищенность от вредоносного ПО.	**Знать:** Программные и программно-аппаратные средства, обеспечивающие реализацию подходов к построению систем, защищенных от вредоносного ПО. **Уметь:** Для конкретной системы осуществлять обоснованный выбор программных и программно-аппаратных средств, обеспечивающих высокую защищенность от вредоносного ПО.
ПК-2 Способен анализировать математические модели систем обеспечения информационной безопасности, а также проводить тестирование средств защиты информации на соответствие этим моделям.	**И-ПК-2_4** Способен применять инструменты статического анализа для изучения принципов работы программ.	**Уметь:** Применять инструменты статического анализа для изучения принципов работы программ.
	И-ПК-2_5 Способен применять инструменты динамического анализа для изучения принципов работы программ.	**Уметь:** Применять инструменты динамического анализа для изучения принципов работы программ.
	И-ПК-2_6 Способен применять сканеры и мониторы для изучения принципов работы программ.	**Уметь:** Применять сканеры и мониторы для изучения принципов работы программ.
	И-ПК-2_7 Способен применять инструменты сигнатурного анализа для выявления вредоносного ПО.	**Уметь:** Применять инструменты сигнатурного анализа для выявления вредоносного ПО.

4. Объем, структура и содержание дисциплины.

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1	Анализ программных реализаций.	2	8	6				12	Задания для самостоятельной работы
2	Защита программ от изучения, несанкционированного копирования и использования.	2	6	4				10	Задания для самостоятельной работы
3	Вредоносное программное обеспечение.	2	6	2				10	Задания для самостоятельной работы
4	Внедрение вредоносного программного обеспечения.	2	6					10	Задания для самостоятельной работы
5	Противодействие вредоносному программному обеспечению.	2	6	2				12	Задания для самостоятельной работы
						2	0,5	33,5	Экзамен
	Всего за 9 семестр		32	16		8			
	Всего	144	**32**	**16**		8	**0,5**	**87,5**	

Содержание разделов дисциплины:

Тема 1. Анализ программных реализаций.

1. Введение.
2. Основны тестирования черного ящика.
3. Мониторинг: доступ к объектам ОС, вызов API ОС, сетевая активность.
4. Основы статического анализа. Ассемблер. Архитектуры x86 и x64. Дизассемблер.
5. Формат исполняемых файлов PE. Windows API. Приложения UEFI.
6. Формат исполняемых файлов ELF. Linux API.

7. Формат исполняемых файлов языков программирования с промежуточным кодом: Java, C#, Python.

8. Декомпиляция. Идентификация абстракций языка C++ при декомпиляции. Виртуальные таблицы. Варианты передачи аргументов и возврата результата из подпрограмм.

9. Динамический анализ. Отладка. Принципы функционирования отладчиков. Отладка в режиме пользователя и в режиме ядра. Снятие дампа.

10. Пошаговая отладка. Программные и аппаратные точки останова.

11. Методика динамического анализа: этапы и методы. Анализ потока данных.

12. Наложение патчей на программы и процессы.

13. Инструментализация. Анализ потока данных. Анализ потока управления. Противодействие антиотладке.

14. Виртуализация. Эмуляция платформы для исследуемой программы. Изоляция исследуемой программы.

Тема 2. Защита программ от изучения.

1. Введение.

2. Основы защиты от дизассемблирования.

3. Обфусцирующие преобразования исходного кода.

4. Обфусцирующие преобразования потока управления. Обфускация библиотечных вызовов, в том числе API ОС. Вызов библиотечных функций по их хеш-значениям.

5. Обфусцирующие преобразования структур данных.

6. Основы антиотладки.

7. Упаковщики. Шифрование выполняемого кода. Самораспаковывающиеся архивы. Оверлеи. Полиморфный код.

8. Техники обнаружения отладчика.

9. Техники обнаружения гипервизора.

10. Основы защиты от снятия дампа.

11. Техники защиты от копирования.

Тема 3. Вредоносное программное обеспечение (ПО).

1. Классификация вредоносного ПО по модели поведения: наблюдатель, перехват и искажение.

2. Виды вредоносного ПО. Вирусы и черви. Троянское ПО. Rootkit, bootkit и backdoor. Вредоносные утилиты: эксплоиты, шеллкод, спуффинговое ПО, спам и флуд ПО, фишинговое ПО, рекламное ПО, программы-вымогатели, ПО для распространения запрещенной информации, DoS, ПО для создания и модификации вирусов.

3. Бинарное вредоносное ПО на Windows.

4. Бинарное вредоносное ПО на Linux.

5. Вредоносное ПО на языках программирования с промежуточным кодом: Java, C#, Python.

6. Вредоносное ПО на языках командных интерпретаторов: batch, PowerShell, bash.

7. Вредоносные ПО в макросах документов.

8. Вредоносное ПО на мобильных утройствах.

9. Bootkit. Расположение в IPL, VBR, MBR, BIOS, UEFI. Secure Boot, Verified Boot, Measured Boot. Intel BootGuard. ARM Trusted Boot Board.

10. Техники распространения вирусного вредоносного ПО.

11. Техники скрытия вредоносного ПО от обнаружения.

Тема 4. Внедрение вредоносного ПО.

1. Предпосылки к внедрению внедрению вредоносного ПО: уязвимости программного обеспечения, уязвимости политики безопасности, человеческий фактор.

2. Методы внедрения вредоносного ПО : маскировка под «безобидное» ПО, подмена, прямое и косвенное ассоциирование.

Тема 5. Противодействие вредоносному ПО.

1. Методы выявления вредоносного ПО: сигнатурное и эвристическое сканирование, контроль целостности, мониторинг информационных потоков, изолированная программная среда, программные ловушки.

2. Принципы построения политики безопасности, обеспечивающей высокую защищенность от вредоносного ПО.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине.

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.

1) Операционные системы.

1. ОС Windows по программе Microsoft Azure Dev Tools for Teaching.
2. ОС Linux, свободно распространяемые дистрибутивы.

2) Гипервизоры и эмуляторы.

1. VirtualBox. Свободно распространяемое ПО. https://www.virtualbox.org/ (дата обращения 19.01.2022)
2. Эмулятор Android. Бесплатное ПО, часть Android Studio. https://developer.android.com/studio/run/emulator (дата обращения 19.01.2022)

3) Шестнадцатеричные редакторы.

1. HxD. Бесплатное ПО. https://mh-nexus.de/en/hxd/ (дата обращения 19.01.2022)
2. Bless. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/bless (дата обращения 19.01.2022)

4) Мониторы.

1. Утилиты пакета Sysinternals. Бесплатное ПО. https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite (дата обращения 19.01.2022)
2. API Monitor. Бесплатное ПО. http://www.rohitab.com/apimonitor (дата обращения 19.01.2022)
3. strace. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/strace (дата обращения 19.01.2022)
4. ltrace. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/ltrace (дата обращения 19.01.2022)
5. auditd. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/auditd (дата обращения 19.01.2022)
6. inotify-tools. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/inotify-tools (дата обращения 19.01.2022)
7. lsof. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/lsof (дата обращения 19.01.2022)
8. tcpdump. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/tcpdump (дата обращения 19.01.2022)
9. Wireshark. Свободно распространяемое ПО. https://www.wireshark.org/ (дата обращения 19.01.2022)

5) Среды разработки.

1. Visual Studio по программе Microsoft Azure Dev Tools for Teaching.
2. Visual Studio Code. Свободно распространяемое ПО. https://code.visualstudio.com/ (дата обращения 19.01.2022)
3. Android Studio. Бесплатное ПО. https://developer.android.com/studio/ (дата обращения 19.01.2022)

6) Дизассемблеры и декомпиляторы.

1. IDA Freeware. Бесплатное ПО. https://hex-rays.com/ida-free/ (дата обращения 19.01.2022)

2. IDA Pro. Лицензия Educational. https://hex-rays.com/ida-pro/ (дата обращения 19.01.2022)

3. Ghidra. Свободно распространяемое ПО. https://github.com/NationalSecurityAgency/ghidra (дата обращения 19.01.2022)

4. Radare2. Свободно распространяемое ПО. https://rada.re/ (дата обращения 19.01.2022)

5. dotPeek. Бесплатное ПО. https://www.jetbrains.com/decompiler/ (дата обращения 19.01.2022)

6. Java Decompiler. Свободно распространяемое ПО. http://java-decompiler.github.io/ (дата обращения 19.01.2022)

7. JADX. Свободно распространяемое ПО. https://github.com/skylot/jadx (дата обращения 19.01.2022)

8. decompyle3 Свободно распространяемое ПО. https://github.com/rocky/python-decompile3 (дата обращения 19.01.2022)

7) Отладчики.

1. IDA Freeware. Бесплатное ПО. https://hex-rays.com/ida-free/ (дата обращения 19.01.2022)

2. IDA Pro. Лицензия Educational. https://hex-rays.com/ida-pro/ (дата обращения 19.01.2022)

3. Radare2. Свободно распространяемое ПО. https://rada.re/ (дата обращения 19.01.2022)

4. gdb. Свободно распространяемое ПО. https://packages.ubuntu.com/impish/gdb (дата обращения 19.01.2022)

5. WinDbg, часть . https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/ (дата обращения 19.01.2022)

6. OllyDbg. Бесплатное ПО. https://www.ollydbg.de/ (дата обращения 19.01.2022)

7. dnSpy. Свободно распространяемое ПО. https://github.com/dnSpy/dnSpy (дата обращения 19.01.2022)

8. Visual Studio Code. Свободно распространяемое ПО. https://code.visualstudio.com/ (дата обращения 19.01.2022)

9. Android Studio. Бесплатное ПО. https://developer.android.com/studio/ (дата обращения 19.01.2022)

8) Фреймворки инструментализации.

1. Pin. Бесплатное ПО. https://www.intel.com/content/www/us/en/developer/articles/tool/pin-a-dynamic-binary-instrumentation-tool.html (дата обращения 19.01.2022)

9) Утилиты обнаружения вредоносного ПО.

1. ClamAV. Свободно распространяемое ПО. https://www.clamav.net/ (дата обращения 19.01.2022)

2. YARA. Свободно распространяемое ПО. https://github.com/VirusTotal/yara (дата обращения 19.01.2022)

10) Отечественные средства защиты информации.

1. ПО Dallas Lock 8.0-К.
2. ПО ViPNet Client 4.
3. ПО ViPNet Administrator 4.5.
4. Программно-аппаратный комплекс (ПАК) ViPNet Coordinator 4.5.
5. ПАК Соболь, версия 4.
6. ПАК Dallas Lock.
7. Аппаратный модуль доверенной загрузки Аккорд.

11) Другие утилиты.

1. Apktool. Свободно распространяемое ПО. https://ibotpeaches.github.io/Apktool/ (дата обращения 19.01.2022)

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины.

1) Основная литература.

1. Проскурин В.Г. Защита программ и данных, 2-е издание, Учебное пособие для студентов учреждений высшего профессионального образования, М., издательский центр «Академия», 2012г. – 208с.

2. Платонов В.В., Программно-аппаратные средства защиты информации учебник для студентов учреждений высшего профессионального образования, М.: Издательский центр «Академия», 2014. – 336с.

2) Дополнительная литература.

1. Касперски, К. Фундаментальные основы хакерства. Искусство дизассемблирования / К. Касперски. — Москва : СОЛОН-Пресс, 2007. — 448 с. — ISBN 5-93455-175-2 . — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/13649 (дата обращения: 26.01.2022)

2. Касперски, Крис. Компьютерные вирусы изнутри и снаружи / К. Касперски. – Санкт-Петербург [и др.]: Питер, 2006. – 527 с. : ил. – ISBN 5-469-00982-3

3. Казарин, О. В. Надежность и безопасность программного обеспечения : учебное пособие для вузов / О. В. Казарин, И. Б. Шубинский. — Москва : Издательство Юрайт, 2022. — 342 с. — (Высшее образование). — ISBN 978-5-534-05142-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/493262 (дата обращения: 28.01.2022)

3) Ресурсы сети Интернет.

1. Денис Юричев. Reverse Engineering для начинающих [Электронный ресурс] // URL: https://beginners.re (дата обращения 19.01.2022)

2. Чернов А. В. Анализ запутывающих преобразований программ [Электронный ресурс] // URL: http://citforum.ru/security/articles/analysis/ (дата обращения 19.01.2022)

3. Журнал «Хакер» [Электронный ресурс] // URL: https://xakep.ru (дата обращения 19.01.2022)

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

− учебные аудитории для проведения занятий лекционного типа, оборудованные персональной компьютерной техникой с установленными средствами визуализации текстов в формате DOC/DOCX, PDF, F2B, файлов изображений, презентаций и мультимедийных файлов, а также – видеопроектором и жалюзи на окнах;

− учебные аудитории для проведения практических занятий: лаборатория программно-аппаратных средств обеспечения информационной безопасности;

− учебные аудитории для проведения групповых и индивидуальных консультаций,

− учебные аудитории для проведения текущего контроля и промежуточной аттестации;

− помещения для самостоятельной работы;

− помещения для хранения и профилактического обслуживания технических средств обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических и лабораторных занятий – списочному составу группы обучающихся.

Автор(ы):

Доцент кафедры компьютерной безопасности и математических методов обработки информации В.Н. Князев

Приложение №1 к рабочей программе дисциплины
«Защита программ и данных»

Фонд оценочных средств для проведения текущей и промежуточной аттестации студентов по дисциплине

1. Типовые контрольные задания и иные материалы, используемые в процессе текущего контроля успеваемости.

Оценка текущей успеваемости выставляется в зависимости от результатов самостоятельной работы студента и его работы на практических занятиях. Задания для самостоятельной работы выдаются учащимся на последнем часе лекционных занятий по теме. Оценка и обсуждение выполненных студентами заданий по самостоятельной работе производится на практическом занятии по данной теме и учитывается на ряду с результатами практических занятий при выставлении оценки текущей успеваемости.

Каждое практическое занятие рассчитано на 2 академических часа. На практическом занятии преподавателем с помощью доступных информационных технологий и собственных подготовленных материалов демонстрируется решение прикладных задач дисциплины. Студенты повторяют ход решения самостоятельно на отдельных устройствах.

Тема 1. Анализ программных реализаций.

1. Задания для самостоятельной работы по теме 1.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_5 в части усвоения знаний необходимых для проведения анализа программных реализаций на практике.

Проверка сформированности УК-1, индикатор И-УК-1_5 в части усвоения знаний необходимых для применения патчей безопасности на программы и работающие процессы на практике.

1.1. Почему задача анализа программных реализаций является актуальной?

1.2. В чем заключаются достоинства и недостатки метода экспериментов с «черным ящиком»?

1.3. Каковы достоинства и недостатки статического метода?

1.4. Какие проблемы возникают при дизассемблировании программных файлов?

1.5. Какие проблемы возникают при изучении аналитиком листингов дизассемблира?

1.6. В каких случаях статический метод анализа программных реализаций является наиболее эффективным?

1.7. Какими основными факторами ограничиваются возможности отладчиков?

1.8. Почему при разных запусках одной и той же программы некоторые буферы могут располагаться по разным адресам оперативной памяти?

1.9. Почему классическая схема применения метода Step-Trace не годится для анализа графических программ Windows?

1.10. Какие проблемы возникают при анализе динамическим методом параллельного кода?

2. Практическое занятие №1. Статический и динамический анализ программ на языках Java, C# и Python.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_5 в части способности применять инструменты статического и динамического анализа для изучения принципов работы программ на языках Java, C# и Python.

Проверка сформированности УК-1, индикатор И-УК-1_5 в части способности применять патчи безопасности на программы и работающие процессы на языках Java, C# и Python.

3. Практическое занятие №2. Статический и динамический анализ программ для Android.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_5 в части способности применять инструменты статического и динамического анализа для изучения принципов работы программ для Andriod.

Проверка сформированности УК-1, индикатор И-УК-1_5 в части способности применять патчи безопасности на программы и работающие процессы Android.

4. Практическое занятие №3. Мониторинг активности процессов Windows и Linux.

Проверка сформированности ПК-2, индикатор И-ПК-2_6 в части способности применять сканеры и мониторы для изучения принципов работы программ.

5. Практическое занятие №4. Статический анализ бинарных исполняемых файлов Windows.

Проверка сформированности ПК-2, индикатор И-ПК-2_4 в части способности применять инструменты статического анализа для изучения принципов работы программ на Windows.

6. Практическое занятие №5. Статический анализ бинарных исполняемых файлов Linux.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 в части способности применять инструменты статического анализа для изучения принципов работы программ на Linux.

7. Практическое занятие №6. Отладка процессов в Windows.

Проверка сформированности ПК-2, индикатор И-ПК-2_5 в части способности применять инструменты динамического анализа для изучения принципов работы программ на Windows.

8. Практическое занятие №7. Отладка процессов в Linux.

Проверка сформированности ПК-2, индикатор И-ПК-2_5 в части способности применять инструменты динамического анализа для изучения принципов работы программ на Linux.

Тема 2. Защита программ от изучения.

1. Задания для самостоятельной работы.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_6 в части усвоения знаний необходимых для проведения анализа программных реализаций на практике.

1.1. Зачем применяется защита кода от анализа?

1.2. Почему в большинстве современных программ защита кода от анализа не применяется?

1.3. Каковы типичные побочные эффекты оснащения программы средствами защиты от анализа?

1.4. В чем заключаются достоинства и недостатки встроенной защиты кода от анализа?

1.5. В чем состоят достоинства и недостатки пристыковочной защиты кода от анализа?

1.6. Как динамическое изменение кода программы затрудняет анализ программы?

1.7. Каким образом полиморфные преобразования кода программы затрудняют анализ программы?

1.8. Каковы типичные побочные эффекты полиморфных преобразований кода?

1.9. Как косвенные вызовы функций в программе затрудняют анализ программы?

1.10. Как вызовы функций через обработчики исключительных ситуаций затрудняют анализ кода программы?

1.11. Как вызовы функций в отдельных потоках затрудняют анализ кода программы?

1.12. Как вызовы функций по таймеру затрудняют анализ кода программы?

1.13. Как нестандартные способы сравнения данных затрудняют анализ кода программы?

1.14. Как динамический импорт системных функций затрудняет анализ кода программы?

1.15. Какие способы искусственного усложнения алгоритмов обработки данных в программе, защищаемой от анализа, вы знаете?

1.16. Как контроль целостности кода программы затрудняет ее анализ под отладчиком?

1.17. Как генерация программой нефатальных исключительных ситуаций затрудняет ее анализ под отладчиком?

2. Практическое занятие №8. Обфускация исходного кода и его анализ.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_6 в части способности применять инструменты статического и динамического анализа, а также мониторы, для изучения принципов работы обфусцированных программ.

3. Практическое занятие №9. Защита исполняемых файлов от изучения.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_6 в части способности применять инструменты статического и динамического анализа, а также мониторы, для изучения принципов работы защищенных от изучения программ.

4. Практическое занятие №10. Изучение защищенных исполняемых файлов.

Проверка сформированности ПК-2, индикаторы И-ПК-2_4 – И-ПК-2_6 в части способности применять инструменты статического и динамического анализа, а также мониторы, для изучения принципов работы защищенных от изучения программ.

Тема 3. Вредоносное программное обеспечение (ПО).

1. Задания для самостоятельной работы.

Проверка сформированности ПК-2, индикатор И-ПК-2_7 в части усвоения знаний необходимых для применения инструментов сигнатурного анализа для выявления вредоносного ПО на практике.

Проверка сформированности ПК-1, индикатор И-ПК-1_1 в части усвоения знаний программных и программно-аппаратные средств, обеспечивающих реализацию подходов к построению систем, защищенных от вредоносного ПО.

1.1. Какой компьютерный вирус причинил наибольший ущерб за всю историю вычислительной техники?

1.2. Какой компьютерный вирус вызвал наиболее масштабную эпидемию за всю историю вычислительной техники?

1.3. Почему написать вирус для Windows сложнее, чем для MS-DOS?

1.4. Почему первые макровирусы так широко распространились?

1.5. Почему люди пишут компьютерные вирусы?

1.6. Почему компьютерный вирус не должен повторно заражать одни и те же объекты?

1.7. В чем состоят достоинства и недостатки онлайн-вирусов по сравнению с почтовыми вирусами?

1.8. Почему большинство онлайн-вирусов функционируют под управлением операционных систем семейства Windows?

2. Практическое занятие №11. Обнаружение вредоносного ПО с помощью ClamAV и YARA.

Проверка сформированности ПК-2, индикатор И-ПК-2_4 в части способности применять инструменты сигнатурного анализа для выявления вредоносного ПО.

Тема 4. Внедрение вредоносного ПО.

1. Задания для самостоятельной работы.

Проверка сформированности ПК-1, индикатор И-ПК-1_4 в части усвоения знаний программных и программно-аппаратные средств, обеспечивающих реализацию подходов к построению систем, защищенных от вредоносного ПО.

1.1. В чем заключается опасность программных закладок?

1.2. Можно ли внедрить программную закладку в адекватно защищенную компьютерную систему?

1.3. Какие типичные уязвимости защиты компьютерных систем вы знаете?

1.4. Как устроен механизм DEP?

1.5. В чем заключается метод маскировки программной закладки под прикладное программное обеспечение?

1.6. В чем состоит основной недостаток метода маскировки программной закладки под прикладное программное обеспечение?

1.7. В чем заключается метод маскировки программной закладки под системное программное обеспечение?

1.8. Каково основное достоинство метода маскировки программной закладки под системное программное обеспечение?

Тема 5. Противодействие вредоносному ПО.

1. Задания для самостоятельной работы.

Проверка сформированности ПК-1, индикатор И-ПК-1_4 в части усвоения знаний программных и программно-аппаратные средств, обеспечивающих реализацию подходов к построению систем, защищенных от вредоносного ПО.

1.1. Является ли задача выявления компьютерного вируса алгоритмически разрешимой в общем случае?

1.2. Как проверить, нет ли в текущем ядре операционной системы уязвимостей, подобных GetAdmin?

1.3. Чем опасно наличие на рабочем столе пользователя окон, обслуживаемых системными процессами?

1.4. В чем заключаются достоинства и недостатки сигнатурного сканирования?

1.5. Каковы достоинства и недостатки эвристического сканирования?

1.6. Какие достоинства и недостатки имеет контроль целостности программного обеспечения?

1.7. Какие достоинства и недостатки имеет контроль целостности конфигурации системы?

1.8. Можно ли обеспечить эффективную антивирусную защиту одними лишь программно-аппаратными средствами?

1.9. Что относится к основным мероприятиям по организационному сопровождению антивирусной защиты?

1.10. О чем должны быть проинструктированы пользователи сети, оснащенной комплексной системой антивирусной защиты?

1.11. Как проверяется адекватность поведения лиц, ответственных за обеспечение антивирусной защиты сети, в случае успешных вирусных атак?

1.12. Как организуется защита от программных закладок ранее неизвестных типов?

1.13. Почему при ручном обнаружении программных закладок не следует немедленно останавливать обнаруженные вредоносные процессы?

1.14. Как определить, все ли обнаруженные вредоносные программы корректно удалены из системы?

2. Практическое занятие №12. Организация изолированной программной среды и аудита в Windows и Linux средствами по умолчанию.

Проверка сформированности ПК-1, индикатор И-ПК-1_4 в части способности для конкретной системы осуществлять обоснованный выбор программных и программно-аппаратных средств, обеспечивающих высокую защищенность от вредоносного ПО.

3. Практическое занятие №13. Организация изолированной программной среды, аудита и доверенной загрузки Windows средствами отечественных средств защиты.

Проверка сформированности ПК-1, индикатор И-ПК-1_4 в части способности для конкретной системы осуществлять обоснованный выбор программных и программно-аппаратных средств, обеспечивающих высокую защищенность от вредоносного ПО.

9. Список вопросов и (или) заданий для проведения промежуточной аттестации.

Список вопросов к экзамену.

1. Анализ программных реализации черного ящика. Мониторинг доступов к объектам ОС, вызовов API ОС, сетевой активности.
2. Статический метод анализа программных реализаций.
3. Декомпиляция бинарных файлов в язык C++.
4. Динамический метод анализа программных реализаций.
5. Анализ потока данных при помощи инструментализации.
6. Анализ потока управления при помощи инструментализации.
7. Противодействие антиотладке при помощи инструментализации.
8. Виртуализация. Эмуляция платформы для исследуемой программы. Изоляция исследуемой программы.
9. Защита программ от дизассемблирования.

10. Обфусцирующие преобразования исходного кода.

11. Обфусцирующие преобразования потока управления. Обфускация библиотечных вызовов, в том числе API ОС.

12. Обфусцирующие преобразования структур данных.

13. Особенности анализа обфусцированных программ.

14. Защита программ от отладки.

15. Упаковщики. Шифрование выполняемого кода. Самораспаковывающиеся архивы. Оверлеи. Полиморфный код.

16. Техники обнаружения отладчика.

17. Техники обнаружения гипервизора.

18. Основы защиты от снятия дампа.

19. Техники защиты от копирования.

20. Особенности анализа программ, защищенных от отладки.

21. Классификация вредоносного ПО по модели поведения: наблюдатель, перехват и искажение.

22. Виды вредоносного ПО. Вирусы и черви. Троянское ПО. Rootkit, bootkit и backdoor. Вредоносные утилиты: эксплоиты, шеллкод, спуффинговое ПО, спам и флуд ПО, фишинговое ПО, рекламное ПО, программы-вымогатели, ПО для распространения запрещенной информации, DoS, ПО для создания и модификации вирусов.

23. Бинарное вредоносное ПО на Windows.

24. Бинарное вредоносное ПО на Linux.

25. Вредоносное ПО на языках программирования с промежуточным кодом: Java, C#, Python.

26. Вредоносное ПО на языках командных интерпретаторов: batch, PowerShell, bash.

27. Вредоносные ПО в макросах документов.

28. Вредоносное ПО на мобильных утройствах.

29. Bootkit. Расположение в IPL, VBR, MBR, BIOS, UEFI. Secure Boot, Verified Boot, Measured Boot. Intel BootGuard. ARM Trusted Boot Board.

30. Техники распространения вирусного вредоносного ПО.

31. Техники скрытия вредоносного ПО от обнаружения.

32. Предпосылки к внедрению программных закладок: уязвимости программного обеспечения, уязвимости политики безопасности, человеческий фактор.

33. Методы внедрения программных закладок: маскировка под безобидное программное обеспечение, подмена, прямое и косвенное ассоциирование.

34. Принципы построения политики безопасности, обеспечивающей высокую защищенность от программных закладок.

35. Методы выявления программных закладок: сигнатурное и эвристическое сканирование. Инструмент YARA.

36. Контроль целостности программного обеспечения, как метод выявления программных закладок.

37. Контроль целостности конфигурации защищаемой системы, как метод выявления программных закладок.

38. Мониторинг информационных потоков, как метод выявления программных закладок.

39. Методы выявления программных закладок, изолированная программная среда, программные ловушки.

40. Построение изолированной программной среды с помощью Windows AppLocker.

41. Построение изолированной программной среды с помощью SELinux.

42. Построение изолированной программной среды на Windows с помощью ПО DallasLock.

43. Защита процесса загрузки ОС с помощью отечественных АМДЗ.

44. Аудит в ОС Windows.

45. Аудит в ОС Linux.

46. Мероприятия по организационному сопровождению антивирусной защиты.

Правила выставления оценки на экзамене.

В экзаменационные билет включается два теоретических вопроса. На подготовку к ответу дается не менее 1часа.

По итогам экзамена выставляется одна из оценок: «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Оценка «Отлично» выставляется студенту, который демонстрирует глубокое понимание теоретической части дисциплины, полные знания об инструментах и техниках их применения для решения прикладных задач защиты программ и данных. Осуществляет межпредметные связи. Дает развернутые, полные и четкие ответы на вопросы экзаменационного билета и дополнительные вопросы, соблюдает логическую последовательность при изложении материала. Грамотно использует терминологию дисциплины.

Оценка «Хорошо» выставляется студенту, ответ которого на экзамене в целом соответствуют указанным выше критериям, но отличается меньшей обстоятельностью, глубиной, обоснованностью и полнотой. В ответе имеют место отдельные неточности (несущественные ошибки), которые исправляются самим студентом после дополнительных и (или) уточняющих вопросов экзаменатора. Оценка «Хорошо» может быть выставлена студенту, имеющему пробелы в знаниях об инструментах и техниках их применения для решения прикладных задач защиты программ и данных, но обладающему глубоким теоретическим пониманием устройства современных компьютерных систем.

Оценка «Удовлетворительно» выставляется студенту, который демонстрирует поверхностное понимание теории, дает недостаточно полные и последовательные ответы на вопросы экзаменационного билета и дополнительные вопросы, но при этом

демонстрирует умение выделить существенные и несущественные признаки и установить причинно-следственные связи. Студент знает и правильно применяет терминологию дисциплины, но допускает ошибки в определении и раскрытии некоторых основных понятий. При аргументации ответа студент не обосновывает свои суждения. На часть дополнительных вопросов студент затрудняется дать ответ или дает неверные ответы. Имеет существенные пробелы в знаниях об инструментах и техниках их применения для решения прикладных задач защиты программ и данных.

Оценка «Неудовлетворительно» выставляется студенту, который демонстрирует разрозненные, бессистемные знания; беспорядочно и неуверенно излагает материал. Не умеет выделять главное и второстепенное, не умеет соединять теоретические положения с практикой, не устанавливает межпредметные связи. Дает неполные ответы, логика и последовательность изложения которых имеют существенные и принципиальные нарушения, в ответах отсутствуют выводы. Дополнительные и уточняющие вопросы экзаменатора не приводят к коррекции ответов студента. На основную часть дополнительных вопросов студент затрудняется дать ответ или дает неверные ответы. Выставляется студенту, не показывающему знаний об инструментарии дисциплины.

Оценка «Неудовлетворительно» выставляется также студенту, который взял экзаменационный билет, но отвечать отказался.

Приложение № 2 к рабочей программе дисциплины «Защита программ и данных»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Защита программ и данных» являются лекции. Это связано с тем, что в основе данная дисциплина находится на стыке между дисциплинами «Техническая защита информации», «Защита в операционных системах», «Основы построения защищенных компьютерных сетей» и «Основы построения защищенных баз данных». Для успешного освоения дисциплины важно углубленное изучение некоторых разделов указанных дисциплин, как в аудитории, так и самостоятельно, в качестве выполняемых в домашних условиях заданий.

Основная цель самостоятельных работ – помочь усвоить теоретические основы и практические методы защиты программ и данных от изучения, несанкционированного использования, от нарушения их целостности и доступности. Для этого необходимо знать и понимать лекционный материал. Поэтому, в процессе изучения дисциплины, рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, представленный в предлагаемой учебной литературе, необходимо дома еще раз прорабатывать и, при необходимости, дополнять информацией, полученной на консультациях, практических занятиях и из рекомендованных ресурсов сети Интернет.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков использования учебной литературы, в течение обучения проводятся мероприятия текущей аттестации в виде самостоятельных работ (в домашних условиях). Варианты заданий выдаются учащимся на последнем часе лекционных занятий по первым пяти темам. Оценка и обсуждение выполненных студентами заданий по самостоятельной работе производится на практических занятиях и учитывается, на ряду с результатами практических занятий, при оценке текущей успеваемости. Также проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце семестра изучения дисциплины студенты сдают экзамен. Экзамен принимается по экзаменационным билетам, каждый из которых включает в себя два теоретических вопроса. Предусмотрена групповая консультация.

Проблемы защиты программ и данных осложняются наличием «багов» и «эксплойтов», заложенных зарубежными производителями аппаратно-программных решений в соответствии со своим законодательством по национальной безопасности. Поэтому, является совершенно необходимым посещение студентами всех аудиторных занятий, где им разъясняется цель и суть неполного представления информации

российским потребителям о западных информационных технологиях и, конкретно, о методах защиты программ и данных. В результате формируется патриотическая позиция, непримиримость к любым аспектам иностранного проникновения к защищаемой в России информации и управления российскими информационными системами, неприемлемость идеологии терроризма, экстремизма.

Для самостоятельной работы студентам, кроме углубленного освоения основной и дополнительной литературы при подготовке к лекционным и практическим занятиям, выполнении заданий для самостоятельной работы, рекомендуется отслеживать регулярно обновляемые материалы государственных стандартов России по управлению безопасности на официальном сайте Росстандарта России (http://www.standard.gost.ru/wps/portal, дата обращения 19.01.2022) в разделе «Уведомления об утверждении национальных стандартов».

Также, рекомендуется использовать ресурсы сети Интернет, указанные в разделе 7 программы.