Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

В. В. Подтопельный

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Учебно-методическое пособие по изучению дисциплины
для студентов специальности
10.05.03 «Информационная безопасность
автоматизированных систем», специализация
«Безопасность открытых информационных систем»

Калининград
Издательство ФГБОУ ВО «КГТУ»
2025

УДК 004.056(076)

Рецензент
кандидат физико-математических наук, доцент, доцент кафедры информационной безопасности ФГБОУ ВО «Калининградский государственный технический университет» Н. Я. Великите

Подтопельный, В. В.
Программно-аппаратные средства защиты информации: учебно-методическое пособие по изучению дисциплины для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем», специализация «Безопасность открытых информационных систем»/ В.В. Подтопельный. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2025. – 91 с.

Учебно-методическое пособие включает в себя рассмотрение теоретических вопросов в области защиты информации по дисциплине «Программно-аппаратные средства защиты информации». В учебно-методическом пособии приведен тематический план изучения дисциплины. Представлены методические указания по изучению дисциплины. Даны рекомендации по подготовке к промежуточной аттестации в форме зачета и экзамена, курсового проекта и по выполнению самостоятельной работы. Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины «Программно-аппаратные средства защиты информации».

Пособие предназначено для студентов 4, 5 курсов специальности 10.05.03 «Информационная безопасность автоматизированных систем».

Табл. 2, список лит. – 29 наименований

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 26 мая 2025 г., протокол № 4

УДК 004.056(076)

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем», специализация «Безопасность открытых информационных систем», изучающих дисциплину «Программно-аппаратные средства защиты информации».

Цель освоения дисциплины: изучение принципов построения систем защиты информации, способов защиты от угроз безопасности в автоматизированных системах.

Осваивается компетенция ОПК-15:

– способен осуществлять администрирование и контроль функционирования средств и систем защиты информации автоматизированных систем, инструментальный мониторинг защищенности автоматизированных систем.

В результате освоения дисциплины обучающийся должен:

знать:

– основные средства и способы обеспечения безопасности информации, принципы построения систем защиты информации;

– программно-аппаратные средства обеспечения защиты информации автоматизированных систем;

– способы реализации угроз безопасности в автоматизированных системах;

уметь:

– проводить выбор и настройку программно-аппаратных средств обеспечения безопасности информации для использования их в составе автоматизированной системы с целью обеспечения требуемого уровня защищенности автоматизированной системы;

владеть:

– обоснования и внедрения перечня сертифицированных и несертифицированных средств защиты информации, необходимых для создания системы защиты информации автоматизированной системы.

Для успешного освоения дисциплины, в соответствии с учебным планом, ей предшествуют «Безопасность вычислительных сетей», «Безопасность систем баз данных», «Безопасность операционных систем.

Далее в пособии представлен тематический план, содержащий перечень изучаемых тем, обязательных лабораторных работ, мероприятий текущей аттестации и отводимое на них аудиторное время (занятия в соответствии с расписанием) и самостоятельную работу. При формировании личного образовательного плана на семестр следует оценивать рекомендуемое время на изучение дисциплины, возможно, вам потребует больше времени на выполнение отдельных заданий или проработку отдельных тем.

В разделе Содержание дисциплины приведены подробные сведения об изучаемых вопросах, по которым вы можете ориентироваться в случае пропуска каких-то занятий, а также методические рекомендации преподавателя для самостоятельной подготовки, каждая тема имеет ссылки на литературу (или иные информационные ресурсы), а также контрольные вопросы для самопроверки.

Раздел «Текущая аттестация» содержит описание обязательных мероприятий контроля самостоятельной работы и усвоения разделов или отдельных тем дисциплины. Далее изложены требования к завершающей аттестации – диф. зачету и экзамену.

Помимо данного пособия, студентам следует использовать материалы, размещенные в соответствующем данной дисциплине разделу ЭИОС, в которые более оперативно вносятся изменения для адаптации дисциплины под конкретную группу.

Программное обеспечение:

Типовое ПО на всех ПК:

1. Операционная система Windows 10 (получаемая по программе Microsoft «OpenValueSubscription»).
2. Офисное приложение MSOfficeStandard 2016 (получаемое по программе Microsoft «OpenValueSubscription»).
3. Kaspersky Endpoint Security.
4. Google Chrome (GNU).
5. Python (GNU/Linux,macOS и Windows).
6. PascalABC.Net.
7. CODESYS.
8. Cisco Packet Tracer (GNU/Linux, macOS и Windows).
9. Oracle VirtualBox 7.1.6 и VirtualBox Extension Pack 7.1.6 for x86_64 hardware.

1. ТЕМАТИЧЕСКИЙ ПЛАН

Таблица 1 – Тематический план

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч.	Объем самостоя-тельной работы, ч.
		Лекции (8-й семестр – 32 ч ауд.)		
1.1	Методы защиты ПО	Тема 1.1 Введение. Основные понятия	4	-
1.2	Методы защиты ПО	Тема 1.2 Методы защиты ПО	4	18
1.3	Методы защиты ПО	Тема 1.3 Подсистемы и модулей системы защиты ПО от несанкционированного использования	4	12
1.4	Методы защиты ПО	Тема 1.4 Методы и средства обратного проектирования.	4	-
1.5	Методы защиты ПО	Тема 1.5 Методы противодействия обратному проектированию	4	-
1.6	Методы защиты ПО	Тема 1.6 Общие методы защиты программ	4	7,85
1.7	Методы защиты ПО	Тема 1.7 Идентификация и аутентификация с использованием технических устройств	4	-
2.1	Защита от разрушающих программных воздействий.	Тема 2.1 Защита от разрушающих программных воздействий	2	-
2.2	Защита от разрушающих программных воздействий	Тема 2.2 Классификация компьютерных вирусов	2	-
		Всего за семестр:	32	**37,85**
		Лекции (9-й семестр – 48 ч ауд.)		
2.3	Защита от разрушающих программных воздействий	Тема 2.3 Программные закладки	4	

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч.	Объем самостоя-тельной работы, ч.
2.4	Защита от разрушающих программных воздействий	Тема 2.4 Особенности функционирования троянских программ	4	20
3.1	Системы защиты информации	Тема 3.1 Особенности систем защиты информации	4	
3.2	Системы защиты информации	Тема 3.2 Контроль целостности	4	
3.3	Системы защиты информации	Тема 3.3 Подсистема управления доступом.	4	
3.4	Системы защиты информации	Тема 3.4 Подсистема регистрации	4	
3.5	Системы защиты информации	Тема 3.5 Криптографическая подсистема СЗИ	4	20
3.6	Системы защиты информации	Тема 3.6 Гарантирование уничтожение.	4	
3.7	Системы защиты информации	Тема 3.7 Системы активного аудита и АПКШ	2	12
		Всего за семестр:	48	52
		Лабораторные занятия (8-й семестр – 32 ч. ауд.)		
1	Методы защиты ПО	Проверка работоспособности средств защиты компьютера от вирусов	2	-
2	Методы защиты ПО	Защита информации с помощью пароля	8	-

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч.	Объем самостоя-тельной работы, ч.
3	Методы защиты ПО	Программирование под HASP с использованием APIфункций	4	-
4	Методы защиты ПО	Изучение функций программы отслеживании обращений к	4	-
		файловой системе		
5	Методы защиты ПО	Исследование моделей защит ПО. Защита от дизассемблеров	4	-
6	Методы защиты ПО	Исследование моделей защит ПО. Изучение средств динамического исследования программ на примере отладчика. Защита от отладчиков	6	-
7	Защита от разрушающих программных воздействий.	Определение жизненно цикла вредоносных программ и извлечение компьютерного вируса средствами антивирусных программ и утилит. Исследование особенностей внедрения вредоносных программного обеспечения	6	-
Всего за семестр:			**32**	
Лабораторные занятия (9-й семестр –**32 ч. ауд.)**				
1	Защита от разрушающих программных воздействий.	Определение специфики работы вредоносного программного обеспечения	2	-
2	Защита от разрушающих программных воздействий.	Обнаружение и извлечение вредоносного программного обеспечения помощью антивирусных программ и утилит	8	-
3	Системы защиты информации	Изучение функций СЗИ. Подсистема управления доступом. Разграничение доступа	8	-

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч.	Объем самостоя-тельной работы, ч.
4	Системы защиты информации	Организация контроля и построение изолированной программной среды средствами СЗИ. Контроль целостности и регистрация событий СЗИ. Идентификация и аутентификация субъекта доступа в СЗИ	6	-
5	Системы защиты информации	Работа с системой анализа защищенности. Применение программ аудита	6	-
6	Системы защиты информации	Использование резервирования	2	-
		Всего за семестр:	**32**	
		Курсовой проект		
2.1	Название первого раздела	Контрольная точка 1. Раздел проекта 1	-	-
3.1	Название третьего раздела	Контрольная точка 2. Раздел проекта 2	-	-
		Оформление проекта. Защита	34,75	-
			34,75	**0**
		Рубежный (текущий) и итоговый контроль		
2.1	Название второго раздела	Контроль 1 (не предусмотрен)	-	-
3.1	Название третьего раздела	Контроль 2 (не предусмотрен)	-	-
		Итоговый контроль (зачет)		
		Итоговый контроль (экзамен)		
		РЭ КА	14 5,4	**0**
		Всего	**198,15**	**89,85**

ИТОГО 288

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И УКАЗАНИЯ К ИЗУЧЕНИЮ

Раздел 1. Методы защиты ПО
Тема 1.1 Введение. Основные понятия

Перечень изучаемых вопросов

1. Уязвимость компьютерных систем.
2. Классификация программных средств защиты.
3. Механизмы защиты.
4. Проблема защиты программного обеспечения от несанкционированного использования.

Методические указания к изучению

Предварительно требуется определить понятие уязвимости. Рассмотреть классификации уязвимостей по областям, по типам и т. п. Требуется рассмотреть стандарт (ГОСТ) по классификациям уязвимостей.

Предварительно требуется определить понятие средство защиты. Рассмотреть классификации средств защиты по областям, по типам и т. п. Требуется рассмотреть руководящие документы по классификациям средств, механизмов и систем защиты.

Рассмотреть проблемы полного и неполного перекрытия угроз средствами защиты информации.

1. Уязвимость компьютерных систем

Уязвимость – это брешь в безопасности системы, позволяющая злоумышленникам нарушать её работу. Эти бреши могут возникать на разных уровнях. Причины:

- Программные ошибки, такие как переполнение буфера.
- Неправильная настройка оборудования или сетевых служб.
- Устаревшие компоненты, которые не были обновлены.
- Человеческий фактор, проявляющийся в неосторожном поведении пользователей.

Примеры уязвимостей:

- CVE-2021-44228 – уязвимость в Apache Log4j.
- Слабые пароли и открытые порты в firewall.

Риски:

- Утечка данных, DDoS-атаки, внедрение вредоносного кода.

Профилактика: Регулярные аудиты безопасности, обновление программного обеспечения и обучение пользователей помогут избежать многих проблем.

2. Классификация программных средств защиты

Программные инструменты можно классифицировать по их назначению. Проактивная защита:

- антивирусы (Kaspersky, Norton) – сканируют файлы и процессы.

– фаерволы (Windows Defender Firewall) – контролируют сетевой трафик.

Реактивная защита:

– системы обнаружения вторжений (IDS) – анализируют подозрительную активность.

– средства анализа логов (SIEM-системы) – выявляют аномалии.

Шифрование данных:

– VeraCrypt – шифрование дисков.

– GPG – защита электронной почты.

Управление доступом: Active Directory – разграничение прав в корпоративных сетях. Критерии выбора: масштаб системы, тип угроз и бюджет – все это важно учитывать при выборе программного обеспечения для защиты.

1. Механизмы защиты. Основные механизмы защиты обеспечивают триаду безопасности (CIA: Confidentiality, Integrity, Availability).

Технические механизмы:

– Шифрование (AES, RSA) – защищает данные от чтения третьими лицами.

– Цифровые подписи — гарантирует целостность и авторство файлов.

– VPN – безопасная передача данных через публичные сети.

Организационные меры:

– Политики безопасности (например, запрет на использование USB-носителей).

– Регламенты резервного копирования.

Аппаратные решения:

– TPM-модули для хранения криптоключей.

– Смарт-карты для двухфакторной аутентификации.

– Пример применения: HTTPS использует шифрование (SSL/TLS) и цифровые сертификаты для защиты веб-сессий.

4. Проблема защиты программного обеспечения от несанкционированного использования

Несанкционированное копирование и взлом лицензий наносят ущерб разработчикам. Способы защиты:

– Лицензионные ключи – привязка к оборудованию или учетной записи.

– Онлайн-активация – проверка легальности через сервер.

– Обфускация – запутывание кода для усложнения декомпиляции.

– Юридические меры – лицензионные соглашения (EULA), судебные иски.

Проблемы:

– Взлом даже сложных систем (например, Denuvo DRM).

– Неудобства для легальных пользователей (частые проверки лицензий).

– Высокая стоимость внедрения защиты.

Тренды:

– Переход на подписки (SaaS-модель).
– Использование блокчейна для контроля лицензий.

При работе над курсовым проектом обратите внимание на:
– описание уязвимостей, которые требуется приводить в первой (теоретической) главе.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1.: Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (с. 4–13).

3. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2.: Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 4–6).

Контрольные вопросы

1. Приведите классификацию уязвимостей компьютерных систем.
2. Приведите классификации программных средств защиты.
3. Охарактеризуйте проблема защиты современного программного обеспечения от несанкционированного использования.
4. Чем отличается уязвимость от эксплойта?
5. Назовите три примера проактивных средств защиты.
6. Как цифровые подписи обеспечивают целостность данных?
7. Какие аппаратные решения повышают безопасность ПО?
8. Почему онлайн-активация эффективнее статичных лицензионных ключей?
9. Какие этические проблемы возникают при использовании DRM-систем?

Тема 1.2 Методы защиты ПО

Перечень изучаемых вопросов

1. Методы защиты ПО от несанкционированного использования.
2. Модульная архитектура технических средств защиты ПО от несанкционированного использования.

Методические указания к изучению

Требуется обратить внимание на архитектуру технических средств защиты ПО от несанкционированного использования, средства, способы защиты ПО от несанкционированного использования.

Система защиты ПО от несанкционированного использования состоит из двух основных частей:

1. Подсистемы внедрения механизмов системы защиты;
2. Внедряемого защитного кода.

Основными требованиями к системе защиты ПО от несанкционированного использования являются следующие:

– система защиты должна выявлять факт несанкционированного запуска программы;

– система защиты должна реагировать на факт несанкционированного запуска программы;

– система защиты должна противостоять возможным атакам злоумышленников, направленных на нейтрализацию системы защиты.

1. Методы защиты ПО от несанкционированного использования

Несанкционированное использование ПО (пиратство, взлом лицензий) наносит ущерб разработчикам и нарушает авторские права. Для борьбы с этим применяются следующие методы:

1.1 Лицензирование. Суть: привязка ПО к пользователю через уникальные ключи или аккаунты. Примеры:

Одноразовые ключи (например, для Windows).

Подписки (Adobe Creative Cloud, Microsoft 365).

Преимущества: Контроль версий, гибкость тарификации.

Недостатки: Взлом ключей, сложность управления для корпораций.

1.2 Обфускация кода. Суть: Запутывание исходного кода для затруднения декомпиляции. Инструменты: ProGuard (для Java), Obfuscator-LLVM (C++).

Плюсы: Усложняет анализ логики ПО.

Минусы: Не защищает от опытных хакеров, замедляет разработку.

1.3 Аппаратная защита. Методы:

USB-донглы (например, Sentinel HASP).

TPM-модули (хранение ключей в защищённой памяти).

Плюсы: высокая стойкость к взлому. Минусы: дополнительные затраты на оборудование, риск потери ключа.

Онлайн-активация. Суть: проверка лицензии через сервер разработчика. Примеры: Игры с DRM (Steam, Denuvo). Преимущества: Динамический

контроль, блокировка пиратских копий. Недостатки: зависимость от интернета, неудобство для пользователей.

DRM-системы (Digital Rights Management). Функции:

Ограничение копирования и модификации контента.

Шифрование данных (например, Netflix для видео).

Проблемы: Конфликты с легальными пользователями, ресурсоёмкость.

1.6 Юридические меры

Инструменты:

EULA (End User License Agreement) — договор с пользователем.

Судебные иски против распространителей пиратских версий.

Эффективность: Зависит от законодательства страны.

2. Модульная архитектура технических средств защиты ПО

Модульная архитектура позволяет гибко настраивать защиту, комбинируя независимые компоненты.

Основные модули. Модуль лицензирования:

Проверяет ключи или подписки.

Интегрируется с серверами активации (например, Elastic License Server).

Модуль шифрования:

Шифрует критичные данные (алгоритмы AES, RSA).

Реализует защиту исполняемых файлов (ASLR, DEP).

Модуль защиты от отладки:

Обнаруживает запуск в среде дебаггера (OllyDbg, IDA Pro).

Блокирует выполнение при выявлении подозрительной активности.

Модуль обновлений:

Автоматически загружает патчи для устранения уязвимостей.

Пример: механизм обновлений в антивирусах.

Аналитический модуль:

Собирает данные о попытках взлома (логи, IP-адреса).

Используется для улучшения защиты.

2.2 Принципы построения

Независимость модулей: Каждый модуль работает автономно, что упрощает замену компонентов.

Гибкость: Возможность добавлять/удалять модули под конкретные задачи (например, отключить онлайн-активацию для офлайн-режима).

A. Совместимость: Интеграция с разными ОС и платформами (Windows, Linux, мобильные системы).

2.3 Примеры решений

A. Denuvo Anti-Tamper: комбинирует шифрование, защиту от отладки и онлайн-проверки.

B. FlexNet Licensing: модульная система для управления лицензиями в корпоративном ПО.

2.4 Преимущества модульного подхода

A. Масштабируемость: легко адаптировать защиту под новые угрозы.

B. Упрощение тестирования: каждый модуль проверяется отдельно.

С. Снижение затрат: не требуется переписывать всю систему при изменении одного компонента.

Современные методы защиты ПО требуют комбинации технических, юридических и организационных мер. Модульная архитектура упрощает создание гибких и адаптивных систем, способных противостоять evolving-угрозам. Однако баланс между защитой и удобством пользователей остаётся ключевым вызовом для разработчиков.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (с. 4 – 13).

3. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 4–6).

Контрольные вопросы

1. Охарактеризуйте методы защиты ПО от несанкционированного использования.
2. Охарактеризуйте модульную архитектуру технических средств защиты ПО от несанкционированного использования
3. Какие методы защиты эффективны против декомпиляции ПО?
4. Почему аппаратные ключи считаются более надёжными, чем программные лицензии?
5. Как модуль защиты от отладки обнаруживает запуск в дебаггере?
6. В чём недостатки DRM-систем для конечных пользователей?
7. Зачем в модульной архитектуре нужен аналитический компонент?

8. Как онлайн-активация предотвращает использование пиратских копий?

Тема 1.3 Подсистемы и модулей системы защиты ПО от несанкционированного использования

Перечень изучаемых вопросов

1. Функционирование подсистем и модулей системы защиты ПО от несанкционированного использования.
2. Электронные ключи.
3. Модель защиты структурным кодом

Методические указания к изучению

Требуется обратить внимание на связь подсистем и модулей системы защиты ПО от несанкционированного использования, типы, архитектуру электронных ключей, способы, позитивные и негативные факторы защиты структурным кодом.

Базовой основой ключей HASP является специализированная заказная микросхема (ASIC – Application Specific Integrated Circuit), имеющая уникальный для каждого ключа алгоритм работы. В процессе своего исполнения защищенная программа опрашивает подключенный к ПК HASP. Если HASP возвращает правильные ответы, работает по требуемому алгоритму и обладает требуемыми эталонными характеристиками, то программа выполняется нормально.

Существует два способа внедрения защитных механизмов в ПО с помощью электронных ключей HASP.

1. HASP API – с помощью API функций. 2. Пакетный режим (HASP Envelope).

Защита структурным кодом (Pattern Code Security – PCS) является средством, значительно повышающим защищенность приложения, защищаемого с помощью электронных ключей HASP.

PCS реализуется в процессе защиты с помощью HASP API. Использование PCS возможно лишь при наличии доступа к исходным текстам защищаемого приложения.

PCS осуществляет последовательность скрытых вызовов процедуры hasp(), не включая эти вызовы в исходный код явно. После каждого вызова процедуры hasp() происходит переключение на следующий скрытый вызов. Если вызов hasp() вдруг удален из защищенной программы, скрытые вызовы не выполняются, а это означает, что кто-то вмешался в работу программы. Тем самым, PCS не дает удалить либо «заклеить» обращения к процедуре hasp().

Преимуществами использования PCS являются:

– скрывание обращения к HASP;

– трассировка вызовов hasp() для шаблонов практически невозможна, так как их нет в исходном коде;

– легче обнаруживается вмешательство извне (если процедуру отключили); если вызов hasp() будет удален, то шаблоны не обновятся, а это значит;

– кто-то вмешался в работу приложения;

– PCS препятствует эмуляции процедуре hasp().

1. Функционирование подсистем и модулей системы защиты ПО

Система защиты ПО от несанкционированного использования состоит из взаимосвязанных подсистем, каждая из которых выполняет специфические задачи.

1.1 Основные подсистемы и их взаимодействие

1. Подсистема лицензирования:

a. Проверяет легитимность пользователя через ключи, подписки или онлайн-активацию.

b. Пример: Steam проверяет лицензию игры через облачный сервер.

2. Подсистема шифрования:

a. Шифрует исполняемые файлы и данные (алгоритмы AES, RSA).

b. Защищает от анализа и модификации.

3. Подсистема защиты от отладки и реверс-инжиниринга:

a. Обнаруживает запуск в дебаггере (например, через проверку процессов).

b. Использует антиотладочные техники (тайм-чеки, проверки целостности кода).

4. Подсистема обновлений:

a. Автоматически загружает патчи для устранения уязвимостей.

b. Пример: обновления антивирусов.

Взаимодействие:

a. При запуске ПО подсистема лицензирования запрашивает проверку ключа.

b. Если проверка пройдена, шифрование активируется для защиты данных.

c. Защита от отладки работает в фоновом режиме, блокируя попытки взлома.

1.2 Пример архитектуры

- Клиентская часть: модули проверки лицензии, шифрования.
- Серверная часть: база данных лицензий, серверы активации.
-

2. Электронные ключи

Электронные ключи – аппаратные или программные инструменты для аутентификации пользователей.

2.1 Типы электронных ключей

1. Аппаратные ключи:

a. USB-донглы (например, SafeNet eToken): хранят криптографические ключи.

b. TPM-модули: встроены в устройство, защищают данные на уровне железа.

c. Программные ключи:

d. Виртуальные токены (Google Authenticator).

e. Лицензионные файлы с цифровой подписью.

2.2 Принцип работы

- При запуске ПО запрашивает ключ.
- Аппаратный ключ генерирует одноразовый пароль (OTP) или хранит постоянный ключ.
- Проверка происходит через сервер или локально (для офлайн-режима).

Преимущества:

- Высокая стойкость к взлому (физический доступ к ключу обязателен).
- Поддержка офлайн-режима (для USB-донглов).

Недостатки:

- Риск потери или повреждения ключа.
- Высокая стоимость внедрения.

Примеры применения:

- Защита корпоративного ПО (Autodesk, MATLAB).
- Банковские системы (транзакции через eToken).

3. Модель защиты структурным кодом

Защита структурным кодом – встраивание механизмов безопасности непосредственно в логику программы.

3.1 Основные методы

1. Обфускация кода:

a. Замена понятных имен переменных на случайные символы.

b. Добавление «мусорного» кода для запутывания логики.

c. Инструменты: ProGuard, Obfuscator-LLVM.

2. Встроенные проверки лицензии:

a. Код, проверяющий наличие ключа, разбросан по всей программе.

b. Пример: функция проверки лицензии вызывается в случайные моменты.

3. Самомодифицирующийся код:

a. Программа изменяет свой код во время выполнения.

b. Затрудняет статический анализ.

3.2 Пример реализации

- Проверка целостности:

```
python
Copy
if verify_license():
    run_program()
else:
    shutdown()
```

- Антиотладочные ловушки:

```
c++
Copy
if(IsDebuggerPresent()){
exit(1);
}
```

Преимущества:

- Усложняет реверс-инжиниринг.
- Не требует внешних модулей.

Недостатки:

- Замедление работы программы.
- Ограниченная эффективность против опытных хакеров.

Заключение

Эффективная защита ПО требует комбинации подсистем, аппаратных ключей и структурных изменений кода. Электронные ключи обеспечивают физический уровень безопасности, а встроенные механизмы защиты усложняют взлом. Однако разработчикам важно балансировать между надёжностью и производительностью, чтобы не ухудшить пользовательский опыт.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (с. 6–84).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Часть 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978-5-7481-0470-8 (6 авт. л.) (с. 6–84).

Контрольные вопросы

1. Функционирование подсистем и модулей системы защиты ПО от несанкционированного использования.
2. Электронные ключи.
3. Модель защиты структурным кодом.
4. Как подсистема защиты от отладки взаимодействует с модулем лицензирования?
5. Чем отличаются аппаратные и программные электронные ключи? Приведите примеры.
6. Почему TPM-модули считаются более безопасными, чем USB-донглы?
7. Какие методы структурной защиты можно применить для противодействия декомпиляции?
8. Как самомодифицирующийся код затрудняет анализ программы?
9. В чём недостатки использования обфускации как основного метода защиты?

Тема 1.4 Методы и средства обратного проектирования

Перечень изучаемых вопросов

1. Методы обратного проектирования.
2. Средства обратного проектирования.

Методические указания к изучению

Классификация и особенности методов и средств атаки на средства защиты программного обеспечения

Наиболее распространенные способы, используемые злоумышленником при реализации первой либо второй угрозы – использование специализированных средств исследования работы программ, а также их кода.

Существует несколько задач, которые злоумышленник должен решить при реализации данных угроз.

1. Задача обнаружения в коде программы модуля защиты. Следует отметить, что без использования специализированных программных средств эта задача в принципе не решаема за приемлемое время. Это обусловлено следующими обстоятельствами.
2. Задача исследования модуля защиты и понимания принципов его действия. Злоумышленник должен понять, каким образом построена защита, где она хранит (если хранит) ключевую информацию, где сохраняет (если сохраняет) свои метки и ключи, на каком этапе принимается решение о регистрации программы, либо об отклонении регистрации.

Специфика атак на модули проверки корректности ключевой информации

Для вскрытия данного типа защит в первую очередь необходимо найти в коде программы код модуля защиты и, а в нем – процедуру данной проверки.

Специфика атак на модули проверки истечения временного срока работы программы или ограничения по количеству ее запусков

Взлом данных модулей во многом практически аналогичен взлому модулей проверки корректности ключевой информации. Специфика взлома здесь состоит в том, что у данных модулей появляются дополнительные уязвимости, которые могут быть использованы злоумышленником.

Отлов злоумышленником вызова WinAPI функций при взломе ПО

Одна из основных задач, которую необходимо решить злоумышленнику при реализации взлома – локализовать модуль защиты в коде программы. Грубая локализация данного модуля решается без существенных затрат с помощью современных средств отладки программного обеспечения. В случае взлома Windows – приложения данная задача решается практически мгновенно путем отслеживания вызовов WinAPI функций, используемых разработчиком.

Средства мониторинга событий – утилиты, отслеживающие операции, производимые программным обеспечением над файлами, реестром, портами, а также отслеживающие потоки системных сообщений.

1. Методы обратного проектирования

Обратное проектирование (reverse engineering) – процесс анализа объекта (ПО, устройства, алгоритма) для восстановления его структуры, логики или технической документации без доступа к исходным данным. Применяется в легальных целях:

- Анализ уязвимостей.
- Совместимость с устаревшими системами.
- Восстановление потерянного кода.
- Изучение алгоритмов конкурентов (в рамках закона).

1.1 Декомпиляция. Суть: Преобразование исполняемого файла (бинарного кода) в код высокоуровневого языка (C, Java). Примеры:

- Java: FernFlower (восстанавливает исходники из .class файлов).
- C/C++: Ghidra, IDA Pro (частичная декомпиляция).
- Ограничения:
- Потеря комментариев и меток.
- Невозможность полного восстановления исходного кода.

1.2 Дизассемблирование. Суть: Преобразование машинного кода в ассемблерные инструкции.

- Использование:
- Понимание логики работы программ на низком уровне.
- Поиск уязвимостей (например, переполнение буфера).
- Инструменты: IDA Pro, Radare2, Binary Ninja.

1.3 Динамический анализ. Суть: Исследование программы во время её выполнения. Методы:

- Отладка: Пошаговое выполнение кода (OllyDbg, GDB).
- Сниффинг памяти: Извлечение данных из оперативной памяти (Cheat Engine).
- Хуки: Перехват системных вызовов (Frida, API Monitor).

1.4 Анализ сетевого трафика. Цель: Изучение взаимодействия ПО с серверами. Инструменты:

- Wireshark (перехват пакетов).
- Burp Suite (анализ веб-запросов).

Пример: Расшифровка протокола обмена данными мобильного приложения.

1.5 Анализ файловой структуры. Суть: Изучение форматов файлов, используемых программой. Инструменты:

- Hex-редакторы (HxD, 010 Editor).
- Специализированные парсеры (Kaitai Struct).

2. Средства обратного проектирования

2.1 Инструменты для дизассемблирования и декомпиляции

- IDA Pro. Особенности: Поддержка множества архитектур (x86, ARM), графическое представление кода.
- Применение: Анализ вредоносного ПО, восстановление алгоритмов.

Ghidra. Плюсы: Бесплатность, встроенный декомпилятор. Минусы: Высокий порог входа для новичков.

Binary Ninja. Имеет удобный API для автоматизации анализа.

2.2 Инструменты для динамического анализа

OllyDbg. Назначение: Отладка Windows-приложений. Пример: Поиск точек проверки лицензии в программе.

Frida. Особенность: внедрение скриптов в работающие процессы. Использование: обход защит мобильных приложений.

x64dbg. Плюсы: Поддержка 64-битных приложений, открытый исходный код.

2.3 Специализированные утилиты

- Wireshark. Функции: Анализ сетевых протоколов, фильтрация трафика.
- Process Monitor. Задача: Мониторинг системных событий (регистр, файлы, сеть).
- PEiD. Цель: Определение компиляторов и упаковщиков исполняемых файлов.

2.4 Современные тренды. ИИ-ассистированный анализ. Пример: Использование нейросетей для предсказания логики обфусцированного кода.

Облачные решения. Платформы: ANY.RUN, Hybrid Analysis (анализ вредоносного ПО в песочнице).

Обратное проектирование – мощный инструмент для анализа ПО, но его использование требует глубоких знаний и соблюдения правовых норм. Современные средства (Ghidra, Frida, Wireshark) значительно упрощают процесс, однако борьба с обфускацией и защитой кода остаётся сложной задачей. Понимание методов reverse engineering критически важно для специалистов по кибербезопасности и разработчиков, стремящихся создавать устойчивые к взлому системы.

Требуется обратить внимание на:

– Методы обратного проектирования, средства обратного проектирования и средства атаки на средства защиты программного обеспечения

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 9–40).

2. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 4–6, 6–10).

Контрольные вопросы

1. Привести методы обратного проектирования.
2. Охарактеризовать средства обратного проектирования.
3. Привести классификацию и особенности методов и средств атаки на средства защиты программного обеспечения
4. Чем отличается декомпиляция от дизассемблирования?
5. Какие инструменты вы бы использовали для анализа сетевого трафика мобильного приложения?
6. Как динамический анализ помогает обнаружить уязвимости типа «переполнение буфера»?
7. В чём преимущество Ghidra перед IDA Pro?
8. Зачем при обратном проектировании применяют хук-инструменты вроде Frida?
9. Какие ограничения есть у декомпиляторов для C++?

Тема 1.5 Методы противодействия обратному проектированию

Перечень изучаемых вопросов

1. Методы противодействия отладчикам защищенного режима.
2. Методы противодействия отладчикам реального режима.

Методические указания к изучению

Требуется рассмотреть трики противодействия отладчикам защищенного режима.

Требуется рассмотреть трики противодействия отладчикам реального режима.

Для вычисления файлов, в которых модуль защиты хранит для себя служебную, ключевую информацию, цифровые подписи, и т. д.

Для вычисления секретных не документируемых файлов, в которых модуль защиты хранит конфиденциальную информацию. Такие файлы иногда используются в слабых программных защитах и, как правило, хранятся во временных либо системных папках.

Для вычисления тех файлов, в которые модуль защиты записывает информацию при установке ПО. Как правило, это бывает необходимо для снятия защит, ограничивающих функционирование во времени использования.

Для вычисления записей в системном реестре Windows, в которых модуль защиты сохраняет служебную информацию при регистрации.

1. Методы противодействия отладчикам в защищённом режиме

Защищённый режим (Protected Mode) современных процессоров предоставляет механизмы управления памятью и привилегиями, что позволяет использовать сложные методы обнаружения отладки.

1.1 Проверка наличия отладчика через API

IsDebuggerPresent() (Windows API): Возвращает флаг, указывающий, подключён ли отладчик к процессу.

NtQueryInformationProcess(): Запрос информации о процессе, включая флаги отладки. Пример кода:

```
cpp
Copy
if (IsDebuggerPresent()) {
    exit(1); // Завершение программы при обнаружении отладчика
}
```

1.2 Использование исключений

INT 3 (Breakpoint): Внедрение «ловушек» – инструкций int 3, которые вызывают исключение при выполнении.

Обработчики SEH (Structured Exception Handling): Мониторинг исключений, характерных для отладки (например, доступ к недопустимой памяти).

1.3 Проверка времени выполнения

Таймеры: Измерение времени между операциями. Отладка замедляет выполнение, что можно обнаружить. Пример:

```
asm
Copy
RDTSC // Чтение счётчика тактов процессора
SUB EAX, EBX // Сравнение с предыдущим значением
CMP EAX, threshold // Если разница превышает порог – отладка
```

1.4 Анти-отладочные трюки в коде

Код-приманка: вставка мусорных инструкций для запутывания анализа.

Самомодифицирующийся код: изменение кода во время выполнения, что мешает статическому анализу.

1.5 Использование аппаратных возможностей

Детектирование аппаратных брейкпоинтов: проверка регистров DR0-DR7 (контроль точек останова).

TPM (Trusted Platform Module): хранение ключей в защищённой памяти, недоступной для отладчика.

2. Методы противодействия отладчикам в реальном режиме

Реальный режим (Real Mode) не поддерживает защиту памяти, поэтому методы противодействия опираются на низкоуровневые техники.

2.1 Перехват прерываний отладчика

Замена векторов прерываний: Например, перехват INT 1 (трассировка) и INT 3 (брейкпоинт).

Пример:

```
asm
Copy
MOV AX, 2501h // Установка своего обработчика для INT 1
MOV DX, offset custom_handler
INT 21h
```

2.2 Проверка состояния системы

Прямой доступ к портам ввода-вывода: отладчики могут эмулировать порты, что можно обнаружить. Чтение/запись в зарезервированные области памяти: Например, область BIOS (0xF000:0xFFFF).

2.3 Таймминговые атаки. Счётчик тактов процессора: использование инструкции RDTSC для детектирования замедления. Циклы задержки: Вставка циклов, время выполнения которых меняется при отладке.

2.4 Использование недокументированных команд

ICE (In-Circuit Emulator) детекция: Недокументированные команды (например, LOADALL), которые не эмулируются отладчиками.

2.5 Самомодифицирующийся код

Динамическое шифрование/расшифровка: Код расшифровывается только во время выполнения, что мешает статическому анализу.

Противодействие отладчикам требует разных подходов в зависимости от режима работы процессора. В защищённом режиме используются API-вызовы, исключения и аппаратные возможности, в реальном – низкоуровневые техники, такие как перехват прерываний и таймминговые проверки. Эффективная защита часто комбинирует несколько методов, чтобы усложнить анализ и взлом ПО.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL:

https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 1).

Контрольные вопросы

1. Охарактеризуйте методы противодействия отладчикам защищенного режима.
2. Охарактеризуйте методы противодействия отладчикам реального режима.
3. Какая инструкция процессора позволяет обнаружить замедление выполнения кода при отладке?
4. Почём замена векторов прерываний эффективна в реальном режиме?
5. Чем отличается обработка аппаратных брейкпоинтов в защищённом и реальном режимах?
6. Как самомодифицирующийся код усложняет работу отладчика?
7. Какие методы подходят для защиты загрузчика, работающего в реальном режиме?

Тема 1.6 Общие методы защиты программ

Перечень изучаемых вопросов

1. Методы противодействия дизассемблированию программного обеспечения.
2. Общие методы защиты программ от отладки и дизассемблирования.

Выделяют несколько общих подходов к защите ПО от дизассемблирования.

1. Шифрование кода. Защищаемый участок кода шифруется каким-либо алгоритмом, а в программу добавляется модуль расшифровки, который в нужный момент расшифровывает его и передает ему управление. В данном случае, защищаемый участок кода перед дизассемблером предстанет в зашифрованном виде, и будет воспринят дизассемблером неверно.
2. Самомодификация кода программой.
3. Различные ходы, приводящие к обману дизассемблера. Этот способ заключается в том, чтобы с помощью различных «хитрых» ходов запутать дизассемблер, подсунув данные вместо кода, или дезориентировать его логику,

повести его по ложному следу. В качестве примеров такой защиты можно привести следующие участки кода.

4. Сокрытие команд передачи управления.

Сокрытие команд передачи управления приводит к тому, что дизассемблер не может построить граф передачи управления. Например, можно модифицировать адреса переходов в ходе выполнения программы (только для реального режима).

5. Использование методики косвенной передачи управления также затрудняет анализ дизассемблированного кода.

6. Использование нестандартных способов передачи управления.

Методы фактора внимания, сокрытие команд передачи управления и метод косвенной передачи, применяемой при противодействии.

Особенности динамического исследования. Метод чёрного ящика, маяков, step-trace. Метод черного ящика:

1) Возможность отслеживания зависимостей на уровне бинарного кода (модификация отдельных байт в заголовках бинарного кода).

2) Исследование функционала криптозащиты:

1. Выявление марканта (случайная последовательность символов) в криптосистеме.

2. Зависимости марканта.

3. Выявление типа криптографического преобразования и т. д. Метод маяков:

a. маяки – это точки программы, действия которых ясны без знания контекста (вызовы динамических библиотек).

b. Все точки останова на все маяки.

c. Установка точки останова на обработки, соответствующие системным вызовам.

d. Метод step-trace 1-го этапа: используется для поиска функций безопасности с учетом внешних проявлений

e. Метод step-trace 2-го этапа: предполагает пошаговый проход от точки останова или маяка при steptrace 1-го этапа

Несмотря на то, что конкретные реализации данных типов защит, зачастую, значительно различаются, можно выделить несколько общих подходов, используемых как в первом, так и во втором типе. Данные подходы представлены ниже.

1. Использование триков (ловушек), с помощью которых можно выявить наличие отладчика в оперативной памяти, и, соответственно, прекратить работу, либо затруднить процесс отладки.

2. Определение наличия отладчика в оперативной памяти используя различные «дырки», допущенные при реализации отладчиков либо внедренные разработчиком отладчика принудительно.

Использование недокументированных команд и возможностей процессора.

Использование того, что некоторые отладчики при загрузке отлаживаемой программы не могут полностью эмулировать «чистую» среду ее запуска в ОС (например, обнуляют некоторые регистры, которые могут нести определенный смысл).

Рассмотрим более подробно реализации защит против отладчиков реального и защищенного режимов.

Особенностью отладчиков защищенного режима является возможность их полной изоляции от выполняемой программы. В связи с этим задача обнаружения отладчика в памяти стандартными средствами значительно усложняется.

Кроме этого, особенностью защищенного режима является введение специализированных отладочных регистров DR0 – DR7, предназначенных для отладочных целей (таких как установка точек останова на обращение к определенным адресам памяти и портам).

Наиболее предпочтительным способом защиты ПО от отладки и дизассемблирования является способ, основанный на шифровании кода программы на некотором секретном ключе. При этом предъявляется требование того, чтобы секретность ключа не могла быть нарушена путем исследования кода программы и дискового пространства ПК.

1. Методы противодействия дизассемблированию

Дизассемблирование – процесс преобразования машинного кода в ассемблерные инструкции. Для защиты от него применяются следующие методы:

1.1 Обфускация кода. Суть: запутывание логики программы для затруднения анализа. Техники:

- переименование переменных и функций (например, func1, var_a вместо осмысленных названий).
- добавление «мусорного» кода: Вставка неисполняемых или бессмысленных инструкций.
- Изменение структуры программы: Разрыв линейного потока выполнения (например, переходы через jmp).

- Инструменты:
- Obfuscator-LLVM (для C/C++).
- ProGuard (для Java/Kotlin).
- Пример кода:

Copy

```
// Исходный код
int checkPassword(char* input) { ... }

// После обфускации
int a1b2(char* x){... jmp label;...}
```

Преимущества: Усложняет статический анализ. Недостатки: Не защищает от динамической отладки.

1.2 Шифрование исполняемого кода

- Суть: Код хранится в зашифрованном виде и расшифровывается только перед выполнением.
- Методы:

a. Упаковщики (ASPack, UPX): Шифруют секции исполняемого файла.

b. Самодельные алгоритмы: Динамическая расшифровка в памяти.

Пример:

asm

Copy

```
section .encrypted
   db 0x12, 0x34, 0x56  ; Зашифрованные данные
```

Плюсы: блокирует статическое дизассемблирование. Минусы: расшифрованный код можно перехватить в памяти.

- Полиморфный код. Суть: Код изменяет свою структуру при каждом запуске. Применение:
  - генерация уникальных инструкций для одинаковой логики.
  - динамическое изменение меток и переходов.

1.4 Виртуализация кода

- Суть: Преобразование исходного кода в набор байткодов, исполняемых виртуальной машиной.
- Инструменты: VMProtect, Themida.
- Преимущества: Крайне сложно восстановить исходную логику.
- Недостатки: Высокие накладные расходы на производительность.

2. Общие методы защиты программ от отладки и дизассемблирования

Комплексная защита включает как технические, так и организационные меры.

2.1 Анти-отладочные техники.

Обнаружение отладчика:

- APIвызовы:IsDebuggerPresent(),CheckRemoteDebuggerPresent()(Windows).
- Проверкафлаговпроцесса: Использование NtQueryInformationProcess длявыявленияфлага ProcessDebugPort.

Таймминговые проверки:

- измерение времени выполнения критичных участков кода.

cpp

Copy

```
auto start = std::chrono::high_resolution_clock::now();
// Критичныйкод
auto end = std::chrono::high_resolution_clock::now();
if(end - start> threshold) exit(1);
```

Обработка исключений:

- Перехват исключений, характерных для отладки (например, EXCEPTION_BREAKPOINT).

2.2 Защита целостности программы

- Контрольные суммы: Проверка целостности файла при запуске.

• Анти-дампинг: Блокировка создания дампов памяти (например, через MiniDumpWriteDump).

2.3 Динамическая защита

• Самомодифицирующийся код: Изменение инструкций во время выполнения.

• Рандомизация адресов: Использование ASLR (Address Space Layout Randomization).

2.4 Организационные меры

• Лицензирование: Привязка к аппаратным ключам (USB-донглы) или онлайн-активации.

• Юридическая защита: Использование EULA и DMCA для преследования нарушителей.

2.5 Комбинирование методов

• Пример:

– Код обфусцирован.
– Исполняемый файл запакован упаковщиком.
– Проверка на отладчик и целостность выполняется при запуске.
– Критичные функции защищены виртуализацией.

Эффективная защита ПО требует сочетания обфускации, шифрования, анти-отладки и организационных мер. Виртуализация и полиморфизм значительно усложняют дизассемблирование, но могут ухудшить производительность. Ключевой принцип – многоуровневая защита, где каждый слой затрудняет анализ, вынуждая злоумышленника преодолевать множество препятствий. Однако важно сохранять баланс: избыточная защита может сделать ПО непригодным для использования.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 40–80).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 1).

Контрольные вопросы

1. Приведите специфику методы противодействия дизассемблированию программного обеспечения, как статическому методу исследования.
2. Приведите общие методы защиты программ от отладки и дизассемблирования.
3. Чем отличается обфускация от шифрования кода?
4. Как таймингговые проверки помогают обнаружить отладку?
5. Какие инструменты подходят для виртуализации кода на C++?
6. Почему полиморфный код сложнее анализировать?
7. Зачем при защите ПО используют несколько слоёв обфускации?

Тема 1.7 Идентификация и аутентификация с использованием технических устройств

Перечень изучаемых вопросов

1. Идентификация и аутентификация пользователей с использованием технических устройств.
2. Идентификация и аутентификация с использованием индивидуальных биометрических характеристик пользователя.

Методические указания к изучению

Обратите внимание на использование технических устройств.

Обратите внимание на то, каким образом производится идентификация и аутентификация с использованием индивидуальных биометрических характеристик пользователя.

При идентификации/аутентификации пользователей с использованием физических устройств, в качестве пользовательского идентификатора используется некоторое техническое устройство, содержащее уникальный идентификационный номер, используемый для решения задач идентификации владельца, а в отдельных случаях и секретную аутентифицирующую информацию, ограничивающую доступ к устройству. Широко распространенными техническими устройствами, используемыми для решения задач идентификации/аутентификации, являются:

- идентификаторы iButton (Touch Memory);
- бесконтактные радиочастотные карты proximity;
- пластиковые карты; ключи e-Token.

В качестве биометрических характеристик, которые могут быть использованы при аутентификации субъекта доступа, достаточно часто применяют следующие:

1) отпечатки пальцев;
2) геометрическая форма рук;
3) узор радужной оболочки и сетчатки глаз;
4) форма и размеры лица;
5) особенности голоса;

6) биомеханические характеристики почерка;

7) биомеханические характеристики «клавиатурного почерка.

Особенностью применения биометрических систем идентификации и аутентификации личности по сравнению с другими классами систем И/АУ являются следующие:

1. Необходимость обучения биометрической системы для конкретных пользователей, зачастую, достаточно длительного.

2. Возможность ошибочных отказов и ошибочных подтверждений при аутентификации пользователей.

3. Необходимость использования специальных технических устройств для чтения биометрических характеристик, как правило, достаточно дорогостоящих (за исключением, быть может, аутентификации по клавиатурному подчерку).

1. Идентификация и аутентификация с использованием технических устройств

Идентификация – процесс определения пользователя через уникальный идентификатор (логин, номер карты).

Аутентификация – подтверждение подлинности пользователя с помощью секретных данных или физических устройств.

1.1 Типы технических устройств

1. Смарт-карты:
- Принцип работы: Карта содержит чип с криптографическим ключом.
- Примеры: банковские карты, пропуски в защищенные зоны.
- Плюсы: Устойчивость к копированию, поддержка офлайн-режима.
- Минусы: Риск утери или повреждения.
2. USB-токены и донглы:
- Функция: Генерируют одноразовые пароли (OTP) или хранят ключи.
- Примеры: YubiKey, Google Titan.
- Преимущества: Компактность, совместимость с двухфакторной аутентификацией (2FA).
3. TPM-модули (Trusted Platform Module):
- Назначение: Встроенный в устройство чип для хранения ключей шифрования.
- Применение: Защита данных на жестком диске (BitLocker в Windows).
4. Мобильные устройства как аутентификаторы:
- Схемы: Приложения-аутентификаторы (Google Authenticator, Microsoft Authenticator).
- Механизм: Генерация временных кодов на основе общего секрета.

1.2 Процесс аутентификации

1. Пользователь вставляет смарт-карту или USB-токен.
2. Система запрашивает PIN-код для разблокировки устройства.
3. Устройство передает криптографический ключ или OTP на сервер.
4. Сервер проверяет подлинность и предоставляет доступ.

Преимущества:

- Высокая защищенность (физический носитель + PIN).
- Устойчивость к фишингу и перехвату паролей.

Недостатки:

- Зависимость от аппаратного носителя.
- Стоимость внедрения и управления устройствами.

2. Идентификация и аутентификация с использованием биометрических характеристик

Биометрия использует уникальные физические или поведенческие признаки пользователя.

2.1 Типы биометрических характеристик

1. Отпечатки пальцев:
- Технология: Оптические или емкостные сканеры.
- Примеры: Разблокировка смартфонов (Apple Touch ID).
2. Распознавание лица:
- Методы: 2D-камеры (менее безопасно) vs 3D-сканирование (Face ID).
- Риски: Обход с помощью фото или маски.
3. Радужная оболочка глаза:
- Точность: Одна из самых надежных технологий.
- Применение: Аэропорты, военные объекты.
4. Голосовая аутентификация:
- Особенность: Анализ частотных характеристик голоса.
- Уязвимости: Запись голоса для обхода.
5. Поведенческая биометрия:
- Примеры: Почерк, ритм набора текста, жесты.

2.2 Принципы работы

1. Регистрация:
- Пользователь предоставляет биометрические данные (например, сканирует палец).
- Система создает цифровой шаблон (не исходные данные).
2. Верификация:
- Сканирование характеристики → сравнение с шаблоном в базе.
- Совпадение → доступ разрешен.

Преимущества:

- Удобство (не нужно запоминать пароль).
- Уникальность биометрических данных.

Недостатки:

- Риск утечки шаблонов (невозможно «сменить» отпечаток).
- Ложные срабатывания/отказы (особенно при плохом качестве сканирования).

2.3 Безопасность и проблемы. Атаки:

- подделка биометрических данных (силиконовые отпечатки, 3D-маски).
- перехват шаблонов (требуется шифрование данных).

Этические вопросы: сбор и хранение биометрии → риски приватности.

Технические устройства (смарт-карты, TPM) и биометрия обеспечивают высокий уровень защиты, но требуют баланса между безопасностью, удобством и приватностью. Комбинирование методов (например, биометрия + аппаратный токен) минимизирует риски, но увеличивает сложность системы. Развитие технологий (искусственный интеллект для анализа поведения, блокчейн для хранения данных) продолжает совершенствовать аутентификацию, делая её одновременно надёжной и пользовательски дружелюбной.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – RL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 40–84).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 2).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 7–84).

Контрольные вопросы

1. Охарактеризуйте идентификацию и аутентификацию пользователей с использованием технических устройств.
2. Охарактеризуйте идентификацию и аутентификацию с использованием индивидуальных биометрических характеристик пользователя.
3. Чем отличается идентификация от аутентификации?
4. Почему TPM-модули считаются более безопасными, чем USB-токены?
5. Какие биометрические методы лучше защищены от подделки?
6. Как работает двухфакторная аутентификация с использованием смартфона?
7. В чём недостатки голосовой аутентификации?
8. Почему поведенческая биометрия сложнее для подделки, чем отпечатки пальцев?

Раздел 2. Защита от разрушающих программных воздействий.

Тема 2.1 Защита от разрушающих программных воздействий

Перечень изучаемых вопросов

1. Модели взаимодействия прикладной программы и РПВ.
2. Компьютерные вирусы как класс РПВ. Защита от РПВ. Изолированная программная среда.

Методические указания к изучению

Рассмотреть модели взаимодействия прикладной программы и РПВ.

Рассмотреть компьютерные вирусы как класс РПВ. Защита от РПВ. Рассмотреть изолированную программную среду.

1. Модели взаимодействия прикладной программы и РПВ

РПВ (Распространённые Программные Вредоносы) – класс вредоносного программного обеспечения, включающий вирусы, черви, трояны и другие угрозы, способные распространяться и воздействовать на системы.

1.1 Механизмы взаимодействия

Прикладные программы могут взаимодействовать с РПВ через:

- Уязвимости в коде: Ошибки в ПО, позволяющие внедрить вредоносный код (например, переполнение буфера).
- Социальную инженерию: Пользователь запускает заражённый файл, маскирующийся под легитимную программу.
- Внешние ресурсы: Загрузка вредоносных скриптов с compromised-сайтов или через рекламные сети (malvertising).

Примеры:

- Макровирусы: Внедряются в документы (Word, Excel) через макросы.
- Файловые вирусы: Заменяют или модифицируют исполняемые файлы (.exe, .dll).

1.2 Этапы заражения

1. Проникновение: РПВ попадает в систему через email-вложения, съёмные носители или сеть.
2. Активация: Запуск вредоносного кода пользователем или автоматически (через уязвимости).
3. Распространение: Копирование себя в другие файлы/системы (сеть, USB-накопители).

2. Компьютерные вирусы как класс РПВ. Защита от РПВ

2.1 Компьютерные вирусы

- Определение: Программы, способные внедряться в другие файлы и реплицироваться.
- Типы:
  - Резидентные: постоянно находятся в памяти.
  - Нерезидентные: активируются только при запуске заражённого файла.
- Примеры:

– Черви. Распространяются по сети без участия пользователя (WannaCry).

– Трояны. Маскируются под полезное ПО для кражи данных.

2.2 Методы защиты от РПВ

Антивирусное ПО:

– Сканирование файлов и процессов в реальном времени (Kaspersky, Norton).

– Сигнатурный анализ (базы известных угроз) + эвристика (обнаружение подозрительного поведения).

1. Брандмауэры: Контроль сетевого трафика для блокировки несанкционированных соединений.

2. Обновления ПО: Устранение уязвимостей через патчи (например, обновления Windows).

3. Обучение пользователей: Противодействие фишингу и социальной инженерии.

Современные технологии:

a. EDR (Endpoint Detection and Response): Мониторинг угроз на уровне устройств.

b. Sandboxing: Анализ подозрительных файлов в изолированной среде.

3. Изолированная программная среда. Изолированные среды предотвращают распространение РПВ, ограничивая их воздействие.

3.1 Технологии изоляции

Виртуализация:

a. Запуск ПО в виртуальных машинах (VMware, VirtualBox).

b. Даже если вирус активируется, основная ОС останется защищённой.

c. Песочницы (Sandbox): изолированное пространство для запуска непроверенных приложений (например, Sandboxie, Windows Sandbox).

d. Контейнеры:

e. Легковесная изоляция процессов (Docker, Kubernetes).

f. Используются в DevOps для безопасного тестирования кода.

3.2 Применение

a. Браузеры: Запуск вкладок в изолированных процессах (Google Chrome).

b. Банковские системы: Выполнение транзакций в защищённых средах.

c. Анализ вредоносного ПО: Изучение вирусов без риска заражения основной системы.

Преимущества:

a. Минимизация ущерба при атаке.

b. Возможность тестирования подозрительного ПО.

Недостатки:

a. Ресурсоёмкость (требует дополнительной памяти/CPU).

b. Ограниченная интеграция с основной системой.

Защита от РПВ требует комплексного подхода: сочетание антивирусных решений, своевременных обновлений и изолированных сред. Вирусы, как

классический представитель РПВ, эволюционируют, но современные технологии (виртуализация, EDR) позволяют эффективно им противодействовать. Использование изолированных сред не только снижает риски заражения, но и обеспечивает безопасность критически важных операций. Однако важно помнить, что даже самая совершенная защита не заменяет бдительности пользователя.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 184–214).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 1).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 3. Поиск и извлечение вредоносных программ в программной среде. – 2020. – 99 с. (с. 7–84).

4. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 4–54).

Контрольные вопросы

1. Охарактеризуйте модели взаимодействия прикладной программы и РПВ.
2. Охарактеризуйте компьютерные вирусы как класс РПВ. Защита от РПВ
3. Поясните принципы построения изолированной программная среда.
4. Чем резидентные вирусы отличаются от нерезидентных?
5. Как эвристический анализ помогает обнаружить неизвестные угрозы?

6. Почему виртуализация считается надёжным методом изоляции?
7. Какие риски остаются при использовании песочницы?
8. Как EDR-системы дополняют традиционные антивирусы?
9. Зачем контейнеры применяют в разработке безопасного ПО?

Тема 2.2 Классификация компьютерных вирусов

Перечень изучаемых вопросов

1. Типы вредоносных программ
2. Типы классификаций вирусных программ

Методические указания к изучению

Рассмотреть различные типологии вирусных и вредоносных программ.

Исследовать методические документы ФСТЭК с указанием типа вредоносных программ

1. Типы вредоносных программ

Вредоносное программное обеспечение (malware) – это программы, созданные для нанесения ущерба, кражи данных или получения несанкционированного доступа к системам. Основные типы:

1.1 Вирусы.

- Описание: внедряются в файлы и распространяются при их запуске. Примеры:

a. ILOVEYOU (2000) – рассылался через email, перезаписывал файлы.

b. Stuxnet – атаковал промышленные системы.

1.2 Черви. Особенность: самостоятельно распространяются по сети без участия пользователя. Пример: WannaCry (2017) – шифровал данные и требовал выкуп, используя уязвимость Windows.

1.3 Трояны. Суть: маскируются под легитимное ПО для кражи данных или скрытой установки других вредоносных программ. Пример: Zeus – банковский троян, перехватывающий пароли.

1.4 Руткиты. Цель: скрывают присутствие злоумышленника в системе, маскируя процессы и файлы. Пример: TDSS – блокировал антивирусы и перехватывал трафик.

1.5 Программы-вымогатели (Ransomware)

- Действие: Шифруют данные и требуют выкуп за расшифровку.
- Пример: NotPetya – уничтожал данные вместо их восстановления после оплаты.

1.6 Шпионское ПО (Spyware)

- Функция: Собирает информацию о пользователе (пароли, банковские данные).
- Пример: Pegasus – шпионил за журналистами и активистами.

1.7 Рекламное ПО (Adware)

- Задача: Показывает навязчивую рекламу, часто перенаправляя на вредоносные сайты.

1.8 Боты и ботнеты

- Принцип: Зараженные устройства объединяются в сеть для DDoS-атак или рассылки спама.
- Пример: Mirai – атаковал DNS-провайдеров через IoT-устройства.

2. Классификации вирусных программ

Вирусы – подкласс вредоносного ПО, специализирующийся на заражении файлов. Их классифицируют по разным критериям:

2.1 По среде обитания

a. Файловые вирусы. Заражают исполняемые файлы (.exe, .dll). Пример: Chernobyl (1998) – стирал данные на жёстком диске.

b. Загрузочные вирусы. Внедряются в загрузочный сектор диска (MBR). Пример: Brain – первый вирус для ПК (1986).

c. Макровирусы. Распространяются через документы с макросами (Word, Excel). Пример: Melissa (1999) – рассылал себя через Outlook.

2.2 По способу заражения

- Резидентные вирусы. Постоянно находятся в оперативной памяти.
- Нерезидентные вирусы. Активируются только при запуске заражённого файла.

2.3 По уровню опасности

1. Безвредные: выводят сообщения или замедляют работу системы.
2. Опасные: повреждают файлы, нарушают работу ОС.
3. Очень опасные: уничтожают данные или аппаратное обеспечение.

2.4 По способу маскировки

- Стелс-вирусы. Скрывают изменения в файлах (перехватывают системные вызовы). Пример: Frodo (1989) – маскировал свой размер.
- Полиморфные вирусы. Меняют свой код при каждом заражении, чтобы избежать обнаружения. Пример: Storm Worm (2007) – использовал шифрование.

2.5 По объекту атаки

- Вирусы для Windows/Linux/macOS: Эксплуатируют уязвимости конкретных ОС.
- Мобильные вирусы: Распространяются через приложения (Android чаще из-за открытости).

Методы защиты от вредоносных программ

1. Антивирусное ПО. Регулярное обновление баз сигнатур.
2. Межсетевые экраны (Firewalls). Блокировка подозрительного трафика.
3. Резервное копирование данных. Защита от программ-вымогателей.
4. Обновление ПО: устранение уязвимостей.
5. Осторожность с вложениями и ссылками. Противодействие фишингу.

Понимание типов вредоносных программ и их классификаций помогает выбирать правильные методы защиты. Вирусы, черви и трояны эволюционируют, но современные технологии (антивирусы с машинным

обучением, песочницы) позволяют минимизировать риски. Важно сочетать технические меры с осторожностью пользователей — это основа кибербезопасности в цифровую эпоху.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 184–214).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калининигр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 1).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 3. Поиск и извлечение вредоносных программ в программной среде. – 2020. – 99 с. (с. 7–84).

4. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 4–54).

Контрольные вопросы

1. Привести типы вредоносных программ
2. Привести типы классификаций вирусных программ
3. Что такое вредоносная программа, и какова её основная цель?
4. Какие основные типы вредоносных программ существуют? Приведите не менее пяти примеров.
5. Чем отличается компьютерный вирус от троянской программы?
6. Какую функцию выполняет ransomware (программа-вымогатель) и как она воздействует на пользователя?

7. Что такое черви (worms) и как они распространяются, в отличие от вирусов?

8. Какие задачи решают шпионские программы (spyware)? Приведите пример их использования.

9. Как работают руткиты (rootkits) и почему их сложно обнаружить?

10. Что такое ботнеты и какую роль в них играют вредоносные программы?

11. Каковы особенности рекламного ПО (adware) и чем оно опасно для пользователя?

12. В чём заключается угроза программ типа «логические бомбы» (logic bombs)?

Тема 2.3 Программные закладки

Перечень изучаемых вопросов

1. Программные закладки, пути их внедрения, методы их выявления.
2. Жизненный цикл вредоносных программ. Структура компьютерных вирусов.

Методические указания к изучению

Рассмотреть отличия программных закладок от вредоносных программ, пути их внедрения, методы их выявления.

Рассмотреть отличия жизненных циклов вредоносных программ. Рассмотреть типовую структуру компьютерных вирусов.

1. Программные закладки: пути их внедрения и методы выявления

Что такое программные закладки?

Программные закладки (бэкдоры) – это скрытые механизмы в программном обеспечении, оборудовании или системах, которые позволяют злоумышленникам обходить стандартные средства аутентификации и получать несанкционированный доступ. Они могут быть внедрены разработчиками (намеренно или случайно) или добавлены позже через уязвимости. Закладки особенно опасны в системах АСУ ТП, так как могут привести к полной компрометации критически важной инфраструктуры.

Пути внедрения программных закладок

1. На этапе разработки:

a. Закладки встраиваются в исходный код программы или прошивку оборудования разработчиками.

b. Пример: в 2013 г. в оборудовании некоторых производителей сетевых устройств были обнаружены предустановленные бэкдоры, позволяющие удалённо получать доступ через стандартные пароли.

2. Через цепочку поставок:

a. Компрометация происходит на этапе производства или доставки оборудования/ПО.

b. Пример: атака через поддельные обновления прошивки для промышленных контроллеров.

3. Эксплуатация уязвимостей:

a. Злоумышленники используют уязвимости в ПО (например, через SQL-инъекции или переполнение буфера) для внедрения закладок.

b. Пример: уязвимость в ПО Siemens WinCC позволяла внедрять код, открывающий доступ к SCADA-системам.

4. Социальная инженерия:

a. Инженеры или администраторы обманом заставляют пользователей установить вредоносное обновление или плагин.

b. Пример: фишинговая атака с поддельным письмом от «техподдержки».

5. Физический доступ:

a. Закладки внедряются путём подключения заражённых USB-устройств или прямого вмешательства в оборудование.

b. Пример: атака Stuxnet, когда вирус распространялся через USB-накопители.

Методы выявления программных закладок

1. Статический анализ кода:

a. Проверка исходного кода на наличие подозрительных функций (например, скрытых точек входа).

b. Инструменты: SonarQube, Checkmarx.

2. Динамический анализ:

- Мониторинг поведения программы в режиме реального времени для выявления аномалий (например, незапланированных сетевых подключений).
- Инструменты: Wireshark, Sysinternals Suite.

3. Сканирование уязвимостей:

- Использование сканеров (Nessus, OpenVAS) для поиска известных сигнатур бэкдоров.

4. Анализ трафика:

- Обнаружение подозрительных исходящих соединений, например, с командными серверами (C&C).
- Пример: системы Nozomi Networks фиксируют нестандартный трафик от PLC.

5. Реверс-инжиниринг:

a. Разборка бинарных файлов или прошивок для поиска скрытых функций.

b. Инструменты: IDA Pro, Ghidra.

6. Тестирование на проникновение:

a. Имитация атак для проверки наличия скрытых точек доступа.

b. Пример: использование Metasploit для поиска открытых портов или слабых паролей.

Пример из практики

В 2021 году на предприятии по производству электроэнергии аудит выявил программную закладку в прошивке ПЛК, которая активировала удалённый доступ при вводе определённой команды через HMI. После обратного проектирования было установлено, что она была добавлена на этапе производства.

2. Жизненный цикл вредоносных программ и структура компьютерных вирусов

Жизненный цикл вредоносных программ

Жизненный цикл вредоносного ПО описывает этапы его существования от создания до нейтрализации. Он включает следующие фазы:

1. Разработка:

a. Злоумышленники пишут код, используя языки программирования (C, Python, ассемблер) или готовые фреймворки (Metasploit).

b. Цель: создание функционала для заражения, маскировки и выполнения задач (например, шифрования данных).

2. Распространение:

a. Вредоносное ПО распространяется через фишинг, заражённые носители, уязвимости в ПО или сети.

b. Пример: WannaCry распространялся через уязвимость EternalBlue в протоколе SMB.

3. Заражение:

a. Программа активируется в целевой системе, внедряясь в файлы, реестр или память.

b. Пример: вирус Stuxnet заражал Windows-системы, а затем передавался на ПЛК через уязвимости Siemens Step 7.

4. Исполнение:

a. Выполняются вредоносные действия: кража данных, шифрование файлов, нарушение работы оборудования.

b. Пример: ransomware Locky шифровал файлы и требовал выкуп.

5. Маскировка:

a. Используются такие техники, как полиморфизм или стелс-механизмы для уклонения от антивирусов.

b. Пример: полиморфные вирусы меняют свой код при каждом запуске.

6. Управление и контроль:

a. Вредоносное ПО может связываться с сервером управления (C&C) для получения команд.

b. Пример: ботнет Mirai координировал атаки через заражённые IoT-устройства.

7. Обнаружение и нейтрализация:

a. Антивирусы, системы IDS/IPS или ручной анализ выявляют и удаляют угрозы.

b. Инструменты: Kaspersky, Malwarebytes, Splunk.

Структура компьютерных вирусов

Компьютерный вирус – это разновидность вредоносного ПО, способная к самокопированию и заражению других программ или систем. Его структура обычно включает следующие компоненты:

1. Модуль заражения (Infection Module):

a. Отвечает за распространение вируса, внедряясь в файлы, загрузочные сектора или сетевые пакеты.

b. Пример: файловый вирус добавляет свой код в исполняемые файлы (.exe).

2. Полезная нагрузка (Payload):

a. Содержит основной вредоносный функционал: удаление данных, шифрование, запуск DDoS-атак.

b. Пример: вирус Chernobyl (CIH) уничтожал данные на жёстком диске и перезаписывал BIOS.

3. Модуль маскировки (Obfuscation Module):

a. Скрывает вирус от обнаружения с помощью шифрования, полиморфизма или маскировки под легитимные процессы.

b. Пример: стелс-вирусы перехватывают системные вызовы, чтобы скрыть своё присутствие.

4. Модуль размножения (Replication Module):

a. Обеспечивает копирование вируса на другие системы или носители.

b. Пример: черви вроде Code Red распространялись через сеть без участия пользователя.

5. Триггер (Trigger):

a. Условие активации полезной нагрузки (дата, действие пользователя, внешняя команда).

b. Пример: «логическая бомба» активируется в заданный день, как вирус Michelangelo (6 марта).

6. Модуль управления (Command Module):

a. Отвечает за связь с сервером злоумышленников или выполнение заранее заданных инструкций.

b. Пример: трояны вроде Zeus собирали банковские данные и отправляли их на C&C-сервер.

Программные закладки и вирусы остаются серьёзной угрозой, требующей многоуровневой защиты. Понимание их жизненного цикла и структуры позволяет разрабатывать эффективные методы противодействия: от статического анализа кода до мониторинга поведения в реальном времени. Однако даже самые совершенные технологии не заменяют осторожности пользователей и регулярного обновления систем.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL:

https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 184–214).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 1).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 3. Поиск и извлечение вредоносных программ в программной среде. – 2020. – 99 с. (с. 7–84).

Контрольные вопросы

1. Привести программные закладки, пути их внедрения, методы их выявления.
2. Привести жизненный цикл вредоносных программ.
3. Привести структура компьютерных вирусов.
4. Какие каналы чаще всего используются для внедрения программных закладок?
5. Как динамический анализ помогает обнаружить закладки?
6. Чем отличается инжектор от репликатора в структуре вируса?
7. Почему этап «сокрытия следов» критичен для жизненного цикла вредоносного ПО?
8. Какие инструменты используются для аудита зависимостей в ПО?
9. Как полиморфизм усложняет обнаружение вирусов?

Тема 2.4 Особенности функционирования троянских программ

Перечень изучаемых вопросов

1. Особенности функционирования троянских программ.
2. Методики распознавания и извлечения вредоносных программ.
3. Классификация методов и средств борьбы с компьютерными вирусами.

Методические указания к изучению

Рассмотреть особенности функционирования троянских программ в режимах ядро и пользовательском режиме.

Рассмотреть методики распознавания и извлечения вредоносных программ: сигнатурный, поведенческий, эвристический анализы.

Рассмотреть классификацию методов и средств борьбы с компьютерными вирусами; способы извлечения вирусов из ОС.

1. Особенности функционирования троянских программ

Троянские программы (трояны) – это разновидность вредоносного ПО, которое маскируется под легитимные приложения или файлы, чтобы обманом заставить пользователя установить его. В отличие от вирусов, трояны не размножаются самостоятельно, но способны выполнять широкий спектр вредоносных действий после активации.

Особенности функционирования:

a. Маскировка. Трояны часто представляются как полезные программы (например, утилиты, игры, обновления) или файлы (PDF, DOC). Пример: троян Emotet распространялся через поддельные письма с вложениями Word.

b. Зависимость от действий пользователя. Активация требует запуска пользователем (например, открытия файла или установки ПО). Пример: троян Zeus распространялся через фишинговые ссылки, которые пользователь должен был открыть.

c. Многофункциональность. После запуска трояны могут выполнять различные задачи: красть данные, устанавливать другие вредоносные программы, создавать бэкдоры. Пример: трояны удалённого доступа (RAT), такие как DarkComet, позволяют удалённо управлять заражённым устройством.

d. Скрытность. Трояны работают в фоновом режиме, избегая явных признаков активности (например, внедряясь в легитимные процессы). Пример: троян TrickBot маскировался под системные службы Windows.

e. Связь с управляющими серверами. Многие трояны подключаются к командным серверам (C&C) для получения инструкций или передачи украденных данных. Пример: троян Dridex отправлял банковские данные на удалённые серверы.

f. Отсутствие самокопирования. В отличие от червей или вирусов, трояны не размножаются, а полагаются на социальную инженерию для распространения. Пример из практики

В 2020 году троян Agent Tesla распространялся через фишинговые письма, выдавая себя за легитимное ПО для логистики. После запуска он собирал пароли, делал скриншоты и отправлял их злоумышленникам, оставаясь незамеченным благодаря шифрованию трафика.

2. Методики распознавания и извлечения вредоносных программ. Распознавание вредоносных программ.

1. Сигнатурный анализ:

a. Сравнение кода подозрительных файлов с базой известных сигнатур вредоносов.

b. Инструменты: антивирусы (Kaspersky, Norton), сканеры (VirusTotal).

c. Преимущество: высокая точность для известных угроз.

d. Недостаток: неэффективен против новых или полиморфных вредоносов.

2. Эвристический анализ:

a. Обнаружение подозрительного поведения (например, попыток шифрования файлов или изменения реестра).

b. Инструменты: ESET NOD32, Comodo.

c. Пример: эвристика выявила программу-вымогатель по массовому изменению расширений файлов.

3. Анализ поведения:

a. Мониторинг действий программы в изолированной среде (sandbox) для выявления аномалий.

b. Инструменты: Cuckoo Sandbox, FireEye.

c. Случай: троян был обнаружен благодаря попытке установить соединение с неизвестным IP-адресом.

4. Сетевой анализ:

a. Проверка трафика на наличие подозрительных соединений или передачи данных.

b. Инструменты: Wireshark, Nozomi Networks.

c. Пример: ботнет выявлен по регулярным запросам к C&C-серверу.

5. Статический анализ кода:

a. Изучение структуры программы без её запуска для поиска вредоносных функций.

b. Инструменты: IDA Pro, Ghidra.

Извлечение вредоносных программ

1. Ручное удаление:

a. Аналитик идентифицирует и удаляет файлы, записи реестра или процессы вручную.

b. Инструменты: Process Explorer, Autoruns.

c. Пример: удаление трояна из автозагрузки Windows.

2. Автоматическое удаление:

a. Антивирусы или специализированное ПО (Malwarebytes, HitmanPro) устраняют угрозу.

b. Преимущество: скорость и доступность.

3. Изоляция в sandbox:

a. Вредоносная программа запускается в виртуальной среде для анализа, после чего удаляется без риска для системы.

b. Инструменты: VMware, VirtualBox.

4. Восстановление системы:

a. Использование резервных копий или точек восстановления для отката изменений.

b. Пример: восстановление после ransomware через образ диска.

5. Декомпиляция и удаление:

a. Реверс-инжиниринг для извлечения и нейтрализации вредоносного кода.

b. Инструменты: OllyDbg, Radare2.

Пример из практики

В 2019 г. программа-вымогатель Ryuk была распознана с помощью эвристического анализа (аномальное шифрование файлов) и удалена с помощью антивируса, после чего система была восстановлена из резервной копии.

3. Классификация методов и средств борьбы с компьютерными вирусами

Методы борьбы с вирусами

1. Профилактика.

Цель: предотвратить заражение.

Подходы:

- Обновление ПО и ОС.
- Ограничение прав пользователей.
- Фильтрация входящего трафика (межсетевые экраны).
  - Пример: блокировка фишинговых писем через почтовый шлюз.

2. Обнаружение:

Цель: идентифицировать угрозу.

Подходы:

- Сигнатурный анализ.
- Эвристический анализ.
- Поведенческий мониторинг.
  - Пример: антивирус фиксирует подозрительный процесс.

3. Нейтрализация:

Цель: устранить вирус.

Подходы:

- Удаление заражённых файлов.
- Карантин подозрительных объектов.
- Прерывание сетевых соединений вредоноса.

Пример: остановка ботнета через блокировку C&C-сервера.

4. Восстановление:

Цель: вернуть систему в рабочее состояние.

Подходы:

- Использование резервных копий.
- Переустановка ОС.
- Ручное удаление следов вируса.
  - Пример: восстановление данных после атаки WannaCry.

Средства борьбы с вирусами

1. Антивирусное ПО:

Функции: сканирование, удаление, защита в реальном времени.

Примеры: Kaspersky, Bitdefender, McAfee.

Особенность: регулярное обновление баз сигнатур.

2. Системы обнаружения вторжений (IDS/IPS):

Функции: мониторинг сети, блокировка атак.

Примеры: Снорт, Суриката.

Применение: защита АСУТП от сетевых вирусов.

3. Сканеры уязвимостей:

Функции: поиск слабых мест, предотвращение эксплуатации.
Примеры: Nessus, OpenVAS.
Пример: обнаружение уязвимостей в SCADA.
4. Песочницы (Sandbox):
Функции: анализ подозрительных файлов в изолированной среде.
Примеры: Cuckoo Sandbox, Any.Run.
Применение: изучение новых угроз.
5. Аппаратные средства:
Функции: фильтрация трафика, изоляция сетей.
Примеры: межсетевые экраны (Cisco ASA), UTM-устройства (FortiGate).
Пример: блокировка червя через сегментацию сети.
6. Специализированные утилиты:
Функции: точечное удаление конкретных вирусов.
Примеры: Kaspersky Virus Removal Tool, Combofix.
Классификация по уровню автоматизации

- Ручные: требуют участия аналитика (реверс-инжиниринг, анализ логов).
- Полуавтоматические: комбинация инструментов и ручной настройки (Wireshark).
- Автоматические: работают без вмешательства (антивирусы, IDS).

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 184–214).

2. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 3. Поиск и извлечение вредоносных программ в программной среде. – 2020. – 99 с. (с. 7–84).

3 Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «»Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3 (с. 11–54).

Контрольные вопросы

1. Привести особенности функционирования троянских программ.
2. Указать и охарактеризовать методики распознавания и извлечения вредоносных программ.
3. Привести классификацию методов и средств борьбы с компьютерными вирусами.

Раздел 3. Системы защиты информации

Тема 3.1 Особенности систем защиты информации

Перечень изучаемых вопросов

1. Особенности организации и функционирования систем защиты информации (СЗИ).
2. Идентификация и аутентификация пользователей СЗИ.

Методические указания к изучению

Особенности организации и функционирования систем защиты информации (СЗИ). Идентификация и аутентификация пользователей СЗИ.

1. Особенности организации и функционирования систем защиты информации

Система защиты информации (СЗИ) – комплекс программных, аппаратных и организационных мер, направленных на обеспечение конфиденциальности, целостности и доступности данных.

1.1 Основные компоненты СЗИ

1. Технические средства:

Межсетевые экраны (Firewalls): Фильтрация трафика для блокировки несанкционированного доступа.

Антивирусное ПО: Обнаружение и нейтрализация вредоносных программ.

Шифрование: Защита данных при передаче и хранении (AES, TLS).

2. Программные модули:

Системы обнаружения вторжений (IDS/IPS): Мониторинг аномальной активности.

DLP-системы (Data Loss Prevention): Предотвращение утечек данных.

3. Организационные меры:

Политики безопасности (например, запрет на использование личных USB-носителей).

Регламенты резервного копирования и восстановления данных.

1.2 Принципы функционирования СЗИ

- Многоуровневая защита (Defense in Depth): Комбинация нескольких слоёв безопасности (сеть, приложения, данные).

- Минимальные привилегии: Пользователи и процессы получают только необходимые права.
- Непрерывный мониторинг: Аудит событий и оперативное реагирование на угрозы.

Пример архитектуры СЗИ:

1. Периметр: Брандмауэр и VPN для защиты сетевого периметра.
2. Уровень доступа: Ролевая модель управления правами (Active Directory).
3. Уровень данных: Шифрование баз данных и использование цифровых подписей.

1.3 Интеграция с инфраструктурой

- Совместимость с облачными сервисами: Защита данных в гибридных средах (AWS, Azure).
- Поддержка IoT-устройств: Аутентификация и шифрование трафика умных устройств.

Сложности:

- Баланс между безопасностью и производительностью.
- Обновление устаревших систем, не поддерживающих современные стандарты.

2. Идентификация и аутентификация пользователей в СЗИ

2.1 Идентификация

Идентификация – процесс распознавания пользователя по уникальному идентификатору:

- Логин (например, user@company.com).
- Смарт-карта или токен (аппаратный идентификатор).

2.2 Аутентификация

Аутентификация – подтверждение подлинности пользователя. Основные методы:

1. Пароли:

Преимущества: Простота внедрения.

Недостатки: Уязвимость к фишингу и брутфорсу.

2. Двухфакторная аутентификация (2FA):

Примеры: SMS-коды, приложения-аутентификаторы (Google Authenticator).

3. Биометрия:

Технологии: Сканирование отпечатков пальцев, распознавание лица.

Использование: Доступ к защищённым серверам или мобильным устройствам.

4. Аппаратные ключи:

Примеры: YubiKey, электронные подписи на базе TPM-модулей.

2.3 Управление учётными записями

- Единая точка входа (SSO): Централизованная аутентификация для всех сервисов компании.

- Ролевой доступ (RBAC): Права назначаются на основе роли (администратор, гость).

Примернастройки RBAC в Active Directory:

```
powershell
Copy
New-ADGroup-Name "Finance_Managers"-GroupScope Global
Add-ADGroupMember-Identity "Finance_Managers"-Members "User1","User2"
```

2.4 Проблемы и решения

- Уязвимости:

Слабые пароли → внедрение политик сложности (минимум 12 символов, буквы+цифры).

Утечка сессий → использование токенов с ограниченным сроком действия.

- Современные тренды:

Zero Trust Architecture: Проверка каждого запроса, даже от авторизованных пользователей.

Адаптивная аутентификация: Усиление проверок при подозрительной активности.

Современные СЗИ требуют интеграции технических, программных и организационных мер. Многоуровневая защита и строгие методы аутентификации (2FA, биометрия) минимизируют риски взломов. Однако эффективность СЗИ зависит не только от технологий, но и от соблюдения политик безопасности пользователями. Внедрение принципов Zero Trust и адаптивной аутентификации становится ключевым трендом, обеспечивающим гибкость и устойчивость к evolving-угрозам

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 40–84).

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калининград. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.) (гл. 2).

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие

для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 7–84).

Контрольные вопросы

1. Особенности организации и функционирования систем защиты информации (СЗИ).
2. Идентификация и аутентификация пользователей СЗИ.

Какие компоненты входят в многоуровневую защиту СЗИ?

3. Чем отличается идентификация от аутентификации?
4. Почему двухфакторная аутентификация безопаснее паролей?
5. Как DLP-системы предотвращают утечки данных?
6. В чём преимущество Zero Trust Architecture перед традиционными моделями?
7. Какие риски возникают при использовании биометрической аутентификации?

Тема 3.2 Контроль целостности

Перечень изучаемых вопросов

1. Контроль целостности и системные вопросы защиты программ и данных
2. Использование СЗИ для контроля целостности. Организация контроля.

Методические указания к изучению

Определить специфику процедур контроля целостности и системные вопросы защиты программ и данных

Рассмотреть использование СЗИ для контроля целостности.

Обратить внимание на прядки организация контроля.

Функциональный контроль предназначен для обеспечения гарантии того, что к моменту завершения загрузки ОС все ключевые компоненты Secret Net 6 загружены и функционируют. Функциональный контроль осуществляется перед входом пользователя в систему.

При функциональном контроле проверяется наличие в системе и работоспособность следующих компонентов:

– ядро;
– модуль входа в систему;
– криптоядро;
– модуль репликации;
– подсистема контроля целостности;
– подсистема аппаратной поддержки.

В случае нарушении функциональной целостности:

– в журнале регистрируется факт нарушения. Это возможно при условии работоспособности ядра.

Администратор информируется об ошибочном завершении функционального контроля.

Вход в систему разрешается только пользователям, входящим в локальную группу администраторов компьютера.

Запуск функционального контроля инициирует модуль входа в систему. При обнаружении нарушений этот модуль управляет административным входом пользователя в систему. Кроме того, он информирует администратора об ошибках контроля.

Если нарушен и сам модуль входа в систему, то при входе пользователя в систему функциональный контроль проводит модуль репликации. Он проверяет, был ли выполнен функциональный контроль, и если нет – инициирует его выполнение.

Процесс инициализации КЦ представляет собой процесс формирования и сохранения списка объектов, подлежащих контролю подсистемой КЦ.

За формирование списка объектов файловой системы (ФС) отвечает приложение для операционных систем Microsoft Windows XP/2003/Vista/2008/7/2008R2 (x86/x64), Linux выполняющее следующие функции:

1. Формирование списка файлов для КЦ;
2. Вычисление контрольных сумм для файлов и секторов HDD, подлежащих КЦ;
3. Формирование объекта (файла), содержащего список объектов контроля и контрольные суммы.

Функции подсистемы обеспечения целостности:

– должна быть обеспечена целостность программных средств СЗИ НСД, обрабатываемой информации, а также неизменность программной среды, при этом:

– целостность СЗИ НСД проверяется при загрузке системы по наличию имен (идентификаторов) компонент СЗИ,

– целостность программной среды обеспечивается отсутствием в

– АС средств разработки и отладки программ;

– должна осуществляться физическая охрана СВТ (устройств и носителей информации), предусматривающая контроль доступа в помещения АС посторонних лиц, наличие надежных препятствий для несанкционированного проникновения в помещения АС и хранилище носителей информации, особенно в нерабочее время;

– должно проводиться периодическое тестирование функций СЗИ НСД при изменении программной среды и персонала АС с помощью тестпрограмм, имитирующих попытки НСД;

– должны быть в наличии средства восстановления СЗИ НСД, предусматривающие ведение двух копий программных средств СЗИ НСД и их периодическое обновление, и контроль работоспособности.

Целостность данных – гарантия того, что информация остается неизменной и не подвергается несанкционированным модификациям. Это одна из ключевых составляющих информационной безопасности наряду с конфиденциальностью и доступностью.

1.1 Методы контроля целостности

1. Хеширование:

Использование алгоритмов (SHA-256, MD5) для генерации уникальных хеш-сумм файлов.

Пример: Проверка целостности дистрибутива ПО через сравнение хеша, указанного разработчиком.

2. Цифровые подписи:

Подписание данных с использованием закрытого ключа для подтверждения их подлинности.

Пример: Электронная подпись документов в системах электронного документооборота.

3. Контрольные суммы (Checksum):

Простые алгоритмы для обнаружения случайных изменений (используются в сетевых протоколах, например, TCP/IP).

1.2 Системные механизмы защиты

- Разграничение прав доступа:

Настройка прав на чтение, запись и выполнение файлов (ACL в Windows, chmod в Linux).

Пример: Запрет модификации системных файлов для обычных пользователей.

- Журналирование файловых систем:

Запись изменений в журнал (например, NTFS в Windows, ext4 в Linux) для восстановления данных после сбоев.

- Защита от вредоносного ПО:

Антивирусы с функцией мониторинга целостности файлов (Kaspersky, Bitdefender).

1.3 Угрозы целостности

- Вредоносное ПО: Вирусы, шифровальщики (Ransomware), трояны.
- Инсайдерские атаки: Умышленное изменение данных сотрудниками.
- Человеческий фактор: Случайное удаление или модификация файлов.

Пример инцидента: Атака NotPetya (2017) шифровала данные и повреждала загрузочные секторы, нарушая целостность систем.

2. Использование СЗИ для контроля целостности. Организация контроля

Системы защиты информации (СЗИ) – комплекс программных, аппаратных и организационных мер для обеспечения безопасности данных.

2.1 Инструменты СЗИ для контроля целостности

1. Системы обнаружения вторжений (IDS/IPS):

Snort, Suricata – анализируют сетевой трафик и файловые изменения на предмет аномалий.

2. DLP-системы (Data Loss Prevention):

Symantec DLP, McAfee DLP – предотвращают утечки и несанкционированное изменение данных.

3. Средства резервного копирования:

Veeam, Acronis – создание копий данных для восстановления после повреждений.

4. Аудит и мониторинг:

SIEM-системы (Splunk, Elastic Security) – сбор и анализ логов для выявления подозрительной активности.

2.2 Организация контроля целостности

1. Оценка рисков:

Определение критичных данных (например, базы данных клиентов, финансовые отчеты).

2. Разработка политик безопасности:

Регламенты резервного копирования, обновления ПО, управления правами доступа.

3. Внедрение технологий:

Настройка автоматической проверки хеш-сумм важных файлов.

Использование EDR-решений (Endpoint Detection and Response) для мониторинга изменений в реальном времени.

4. Обучение персонала:

Тренинги по кибергигиене, предотвращению фишинга и работе с данными.

5. Регулярный аудит:

Проверка соответствия стандартам (ISO 27001, ГОСТ Р 57580) и устранение уязвимостей.

2.3 Пример внедрения СЗИ

- Сценарий: Компания внедрила Tripwire для мониторинга целостности файлов.

Действия:

1. Установка агентов на серверы.
2. Настройка базовых конфигураций (белые списки файлов).
3. Ежедневные отчеты о изменениях.

- Результат: Снижение числа инцидентов, связанных с несанкционированными изменениями, на 70 %.

2.4 Современные тренды

- Искусственный интеллект:

Алгоритмы машинного обучения для обнаружения аномалий (например, нестандартные изменения в логах).

- Блокчейн:

Фиксация транзакций в неизменяемом реестре для обеспечения целостности данных.

Контроль целостности данных – неотъемлемая часть защиты информации. Использование СЗИ, таких как IDS, DLP и SIEM, позволяет

автоматизировать процессы обнаружения и предотвращения несанкционированных изменений. Однако эффективность этих систем зависит от грамотной организации: оценки рисков, разработки политик и обучения сотрудников. Современные технологии, включая ИИ и блокчейн, открывают новые возможности для обеспечения неизменности данных, но их внедрение требует адаптации инфраструктуры и соблюдения стандартов безопасности.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 23–26, 29–94).

Контрольные вопросы

1. Каков контроль целостности программ и данных
2. Чем определяется специфика использования СЗИ для контроля целостности.
3. Какие алгоритмы хеширования наиболее надежны для проверки целостности?
4. Как цифровая подпись связана с контролем целостности?
5. Чем отличается IDS от IPS в контексте защиты данных?
6. Какие этапы включает организация контроля целостности в компании?
7. Как блокчейн может быть использован для обеспечения целостности?
8. Почему инсайдерские угрозы сложнее обнаружить, чем внешние атаки?

Тема 3.3 Подсистема управления доступом

Перечень изучаемых вопросов

1. Разграничение доступа.
2. Управление политикой безопасности.

Методические указания к изучению

Рассмотреть разграничение доступа мандатного типа.

Рассмотреть разграничение доступа дискреционного типа.

Определить специфику управления политикой безопасности МБ, МБС, МО.

1. Разграничение доступа в системах защиты информации (СЗИ)

Разграничение доступа – механизм, обеспечивающий предоставление пользователям и процессам прав доступа только к тем ресурсам, которые необходимы для выполнения их задач. Это ключевой элемент предотвращения утечек данных и несанкционированных действий.

1.1 Основные модели разграничения доступа

1. Дискреционное управление доступом (DAC – Discretionary Access Control):

Принцип: владелец ресурса самостоятельно назначает права доступа (чтение, запись, выполнение).

Пример: файловая система NTFS в Windows, где пользователь устанавливает права для папок/файлов через ACL (Access Control List).

Плюсы: гибкость.

Минусы: риск ошибок из-за человеческого фактора.

2. Мандатное управление доступом (MAC – Mandatory Access Control):

Принцип: права определяются системой на основе меток безопасности (уровни секретности: «совершенно секретно», «конфиденциально»).

Пример: SELinux в Linux, где доступ к процессам и файлам контролируется политиками безопасности.

Плюсы: высокая защищенность.

Минусы: сложность настройки.

3. Ролевое управление доступом (RBAC – Role-Based Access Control):

Принцип: Права назначаются ролям, а не пользователям. Например, «администратор», «менеджер», «гость».

Пример: Корпоративные системы на базе Active Directory, где доступ к данным зависит от должности сотрудника.

Плюсы: Упрощение управления правами в крупных организациях.

4. Атрибутное управление доступом (ABAC – Attribute-Based Access Control):

Принцип: Доступ определяется динамически на основе атрибутов пользователя, ресурса и контекста (например, время суток, местоположение).

Пример: Облачные сервисы AWS, где политики IAM регулируют доступ к ресурсам.

1.2 Технологии реализации

- ACL (Access Control List): Списки прав для объектов (файлов, папок, устройств).
- Системы аутентификации: Интеграция с LDAP, Active Directory, OAuth 2.0.
- Шифрование: Ограничение доступа через ключи (например, доступ к зашифрованным данным только при наличии сертификата).

Пример:

В банковской системе сотрудник отдела кредитования (роль «кредитный менеджер») имеет доступ только к данным клиентов, но не к финансовым отчетам.

2. Управление политикой безопасности в СЗИ

Политика безопасности – набор правил и процедур, определяющих, как организация защищает свои информационные активы. Управление политикой включает её разработку, внедрение, мониторинг и обновление.

2.1 Этапы управления политикой безопасности

1. Анализ рисков:

Выявление критичных активов (базы данных, серверы) и угроз (утечки, DDoS-атаки).

Инструменты: SWOT-анализ, методологии OCTAVE, NIST SP 800-30.

2. Разработка политик:

Создание документов, регламентирующих:

- Правила доступа.
- Использование шифрования.
- Действия при инцидентах.

Пример: Политика паролей (минимальная длина – 12 символов, обязательная смена каждые 90 дней).

3. Внедрение:

Настройка технических средств (фаерволы, DLP-системы).

Обучение сотрудников.

4. Мониторинг и аудит:

Проверка соблюдения политик с помощью SIEM-систем (Splunk, IBM QRadar).

Регулярные penetration-тесты.

5. Обновление:

Адаптация политик к новым угрозам (например, учет рисков IoT-устройств).

2.2 Инструменты управления политикой безопасности

- IAM-системы (Identity and Access Management):

Управление ролями и правами (AWS IAM, Microsoft Azure AD).

- PAM-решения (Privileged Access Management):

Контроль доступа к критичным системам (CyberArk, Thycotic).

- Автоматизация:

Использование скриптов для массового назначения прав (например, PowerShell для Active Directory).

Пример политики:

В компании внедрена политика «нулевого доверия» (Zero Trust):

- Каждый запрос проверяется, даже если он исходит из внутренней сети.
- Многофакторная аутентификация обязательна для всех сотрудников.

2.3 Стандарты и регуляторика

- ISO 27001: Международный стандарт управления информационной безопасностью.
- GDPR: Регламент защиты персональных данных в ЕС.
- ФЗ-152 (Россия): Закон о защите персональных данных.

Разграничение доступа и управление политикой безопасности – основа защиты информации в организациях. Использование моделей DAC, MAC, RBAC и ABAC позволяет гибко контролировать права пользователей, а современные инструменты (IAM, PAM) автоматизируют процессы. Однако эффективность СЗИ зависит не только от технологий, но и от грамотной разработки политик, их своевременного обновления и обучения сотрудников. Внедрение стандартов (ISO 27001) и принципов Zero Trust обеспечивает соответствие регуляторным требованиям и снижает риски кибератак.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 29–94).

Контрольные вопросы

1. Привести правила разграничения доступа.
2. Описать механизм управления политикой безопасности МБ, МБС, МО
3. Чем отличается дискреционная модель доступа от мандатной?
4. Какие преимущества дает ролевое управление доступом (RBAC) в крупных компаниях?
5. Как атрибутное управление доступом (ABAC) учитывает контекст при принятии решений?
6. Какие этапы включает разработка политики безопасности?
7. Почему стандарт ISO 27001 важен для управления политиками?
8. Как Zero Trust Architecture влияет на разграничение доступа?

Тема 3.4 Подсистема регистрации

Перечень изучаемых вопросов

1. Рассмотреть особенности подсистема регистрации и учета централизованного и децентрализованного типа.

2. Рассмотреть особенности Регистрация и учет событий защищаемой среды.

Методические указания к изучению

Привести особенности подсистемы регистрации и учета.

Привести особенности регистрации и учет событий защищаемой среды.

1. Особенности подсистем регистрации и учёта централизованного и децентрализованного типа

Регистрация и учёт – процессы сбора, хранения и анализа данных о событиях в информационной системе. Эти подсистемы обеспечивают прозрачность, безопасность и соответствие регуляторным требованиям.

1.1 Централизованная подсистема

Особенности:

- Все данные стекаются в единое хранилище (сервер, облако).
- Управление и анализ осуществляются централизованно.

Преимущества:

- Единая точка контроля: Упрощение аудита и корреляции событий.
- Масштабируемость анализа: Возможность использовать мощные инструменты (SIEM-системы, Big Data).
- Снижение риска потери данных: Резервное копирование и шифрование централизованного хранилища.

Недостатки:

- Уязвимость к атакам: Центральный сервер – критичная точка отказа.
- Задержки передачи данных: Проблемы в сетевой инфраструктуре влияют на сбор логов.
- Высокая стоимость: Требуются ресурсы для обработки больших объёмов данных.

Примеры решений:

- SIEM-системы: Splunk, IBM QRadar, ELK Stack (Elasticsearch, Logstash, Kibana).
- Облачные сервисы: AWS CloudTrail, Azure Monitor.

1.2 Децентрализованная подсистема

Особенности:

- Данные хранятся локально на узлах (устройствах, серверах).
- Анализ может выполняться распределённо или агрегироваться периодически.

Преимущества:

- Устойчивость к отказам: Нет единой точки отказа.

- Снижение сетевой нагрузки: Локальное хранение и предобработка данных.
- Конфиденциальность: Чувствительные данные не передаются за пределы узла.

Недостатки:

- Сложность анализа: Требуются инструменты для агрегации данных из разных источников.
- Риск фрагментации: Данные могут быть неполными или противоречивыми.

Примеры решений:

- Распределённые базы данных: Apache Cassandra, Amazon DynamoDB.
- Блокчейн: Hyperledger Fabric (неизменяемость логов).

Таблица 1 – Сравнение подходов

Критерий	Централизованная	Децентрализованная
Управление	Проще	Сложнее
Надёжность	Зависит от сервера	Выше (нет единой точки отказа)
Масштабируемость	Высокая (для больших данных)	Зависит от узлов
Соответствие GDPR	Требует защиты центра	Локальное хранение упрощает

2. Регистрация и учёт событий защищаемой среды

2.1 Типы регистрируемых событий

1. События безопасности:

Попытки несанкционированного доступа.

Изменение прав пользователей.

Срабатывание антивирусных систем.

2. Системные события:

Перезагрузка серверов.

Обновление ПО.

3. Пользовательские действия:

Вход/выход из системы.

Доступ к конфиденциальным файлам.

2.2 Методы регистрации

- Лог-файлы: Текстовые файлы с записями о событиях (например, `/var/log/auth.log` в Linux).
- Системы мониторинга: Prometheus, Nagios (сбор метрик в реальном времени).
- Агенты и датчики: Утилиты (Fluentd, Logstash), собирающие данные с узлов.

2.3 Организация учёта событий

1. Сбор данных:

Режим реального времени: SIEM-системы (Splunk).

Пакетный режим: Периодическая выгрузка логов.

2. Хранение:

Структурированные форматы: JSON, XML (удобство анализа).

Шифрование и контроль доступа: Защита от несанкционированных изменений.

3. Анализ:

Корреляция событий: Выявление паттернов атак (например, множественные попытки входа).

Генерация отчётов: Для аудита и соответствия стандартам (ISO 27001).

2.4 Проблемы и решения

- Большие объёмы данных:

Использование алгоритмов сжатия.

Фильтрация нерелевантных событий.

- Безопасность логов:

Цифровые подписи для предотвращения изменений.

Хранение в неизменяемом хранилище (WORM – Write Once Read Many).

- Ложные срабатывания:

Настройка правил фильтрации и машинное обучение для снижения шума.

Пример: в банковской системе каждое изменение в базе данных клиентов регистрируется с указанием:

- Времени события.
- Идентификатора пользователя.
- Типа операции (создание, изменение, удаление).

Данные шифруются и отправляются в SIEM для анализа.

Выбор между централизованным и децентрализованным подходами зависит от требований к безопасности, масштаба системы и регуляторных норм. Централизованные SIEM-решения подходят для крупных организаций, нуждающихся в глубоком анализе данных, тогда как децентрализованные системы актуальны для распределённых сред с высокими требованиями к отказоустойчивости. Регистрация событий защищаемой среды должна охватывать все критические операции, а безопасное хранение и анализ логов — стать основой для предотвращения инцидентов и доказательства соответствия стандартам.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 9–40).

2. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с .96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 29–94).

Контрольные вопросы

1. Укажите особенности подсистемы регистрации и учета.
2. Приведите особенности Регистрация и учет событий защищаемой среды.
3. Какие преимущества централизованной подсистемы важны для финансовых организаций?
4. Как децентрализованный подход помогает соблюдать GDPR?
5. Какие события обязательно регистрировать в защищаемой среде по стандарту ISO 27001?
6. Чем отличается SIEM от систем пакетного сбора логов?
7. Почему блокчейн используют для децентрализованного учёта событий?
8. Как защитить логи от подделки?

Тема 3.5 Криптографическая подсистема СЗИ

Перечень изучаемых вопросов

Особенности криптографических подсистем СЗИ.

Шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах.

Методические указания к изучению

Рассмотреть виды криптографических подсистем СЗИ.

Рассмотреть шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах

1. Особенности криптографических подсистем в СЗИ

Криптографические подсистемы – ключевой элемент систем защиты информации (СЗИ), обеспечивающий конфиденциальность, целостность и аутентификацию данных. Их основные функции:

1. Шифрование данных: Преобразование информации в нечитаемый формат для защиты от несанкционированного доступа.
2. Управление ключами: Генерация, хранение, распределение и обновление криптографических ключей.
3. Цифровые подписи: Гарантия подлинности и неизменности данных.
4. Аутентификация: Подтверждение личности пользователей и устройств.

Особенности современных криптоподсистем:

- Интеграция с инфраструктурой: Совместимость с облачными сервисами, IoT-устройствами и распределёнными системами.
- Поддержка стандартов: AES, RSA, ECC (Elliptic Curve Cryptography), ГОСТ 34.12-2015.
- Аппаратное ускорение: Использование TPM-модулей, HSM (Hardware Security Modules) для защиты ключей.
- Адаптивность: Возможность обновления алгоритмов для противодействия новым угрозам (например, квантовым атакам).

2. Шифрование информации для различных субъектов доступа

В системах с множеством пользователей или групп требуется гибкое управление доступом через шифрование на разных ключах.

2.1 Методы разделения ключей

1. Симметричное шифрование:

Принцип: Один ключ для шифрования и расшифровки.

Использование: Каждая группа получает уникальный ключ.

Пример: AES-256 для шифрования данных отдела финансов.

Проблема: Сложность безопасной передачи ключей между пользователями.

2. Асимметричное шифрование:

Принцип: Пара ключей (публичный и приватный).

Сценарий:

- Данные шифруются публичным ключом группы.
- Расшифровка возможна только приватным ключом группы.

Пример: RSA для защиты документов проектной команды.

3. Иерархическое шифрование:

Принцип: Ключи организованы в иерархию (например, корпоративный мастер-ключ → ключи отделов → ключи сотрудников).

Пример: Генеральный директор имеет доступ ко всем данным, менеджер – только к своему отделу.

4. Групповые схемы:

Технологии:

- Threshold Cryptography: Для расшифровки требуется участие нескольких пользователей (например, 3 из 5).
- ABE (Attribute-Based Encryption): Доступ определяется атрибутами (роль, должность).

Пример: Медкарты пациентов доступны только врачам определённого отделения.

2.2 Управление ключами

- PKI (Public Key Infrastructure):

Центры сертификации (CA) выпускают и проверяют ключи.

Пример: SSL-сертификаты для веб-ресурсов.

- KMS (Key Management Service):

Облачные решения (AWS KMS, Google Cloud KMS) для централизованного управления ключами.

- Ротация ключей:

Автоматическая смена ключей каждые 90 дней для снижения риска компрометации.

2.3 Примеры использования

- Корпоративная среда:

Шифрование данных R&D-отдела на отдельном ключе, недоступном другим подразделениям.

- Государственные системы:

Многоуровневое шифрование документов с доступом по уровню допуска (совершенно секретно, секретно).

- Медицина:

ABE для доступа к истории болезни на основе роли врача и отделения.

3. Преимущества и сложности

Преимущества:

- Гранулярный контроль: Точная настройка прав доступа для пользователей и групп.
- Снижение риска утечек: Даже при компрометации одного ключа остальные данные защищены.
- Соответствие регуляторным требованиям: GDPR, HIPAA, ФЗ-152.

Сложности:

- Управление большим числом ключей: Требует автоматизации (KMS, HSM).
- Производительность: Асимметричное шифрование ресурсоёмко для больших данных.
- Баланс безопасности и удобства: Слишком сложные системы могут замедлить рабочие процессы.

Криптографические подсистемы СЗИ обеспечивают безопасность данных за счёт гибкого управления ключами и адаптивных алгоритмов. Шифрование для различных субъектов доступа позволяет реализовать принцип минимальных привилегий, минимизируя риски утечек. Однако эффективность таких систем зависит от грамотной реализации управления ключами и баланса между безопасностью и удобством. Современные технологии (ABE, облачные KMS) и стандарты (AES, ГОСТ) делают этот подход доступным для организаций любого масштаба.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. — Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890- 1878-1. – Текст : электронный (с. 84–132)

3. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие

для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: С. 96–97. – ISBN 978- 5-7481-0470-8 (6 авт. л.) (с. 29–94).

Контрольные вопросы

1. Привести особенности криптографических подсистем СЗИ.
2. Привести шифрование информации, принадлежащей различным субъектам доступа (группам субъектов) на разных ключах
3. Чем иерархическое шифрование отличается от группового?
4. Как ABE обеспечивает доступ на основе атрибутов?
5. Почему симметричное шифрование редко используют для разделения доступа между группами?
6. Какие задачи решает PKI в управлении ключами?
7. Как ротация ключей повышает безопасность?
8. Какие проблемы возникают при использовании пороговых схем (Threshold Cryptography)?

Тема 3.6 Гарантирование уничтожение

Перечень изучаемых вопросов

1. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ.
2. Очистка (обнуление, обезличивание) внешних накопителей.

Методические указания к изучению

Рассмотреть процедуры обнуления, обезличивания освобождаемых областей оперативной памяти ЭВМ.

Рассмотреть процедуры очистки (обнуление, обезличивание) внешних накопителей.

1. Очистка (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ

Оперативная память (ОЗУ) хранит данные временно, но при определенных условиях информация может быть восстановлена, особенно если компьютер не выключается полностью. Это критично для систем, работающих с конфиденциальными данными.

1.1 Методы очистки оперативной памяти

Принудительное обнуление (Zeroization):

Суть: заполнение ячеек памяти нулями или случайными значениями перед освобождением.

Листингг «Использование функций memset_s в языке C для безопасной перезаписи данных».

c

Copy

voidsecure_erase(void* ptr, size_tsize) {
memset_s(ptr, size, 0, size);
}

1. Перезагрузка с очисткой кэша. Полное выключение компьютера стирает данные ОЗУ. Однако в режимах гибернации или сна данные сохраняются на диске.

Рекомендация: Отключить гибернацию для конфиденциальных систем.

Использование аппаратных решений. TPM-модули (Trusted Platform Module): Автоматическая очистка памяти при обнаружении несанкционированного доступа. Программные инструменты. RAM cleaners: Утилиты, такие как CleanMem, принудительно освобождают неиспользуемую память.

1.2 Особенности и риски

- Остаточные данные: В кэше процессора или буферах ввода-вывода могут сохраняться фрагменты информации.
- Атаки типа Cold Boot: Злоумышленники могут восстановить данные из ОЗУ, если быстро заморозят память и перенесут её в другое устройство.
- Решение: Шифрование оперативной памяти (TRIM, Intel SGX).

2. Очистка (обнуление, обезличивание) внешних накопителей

Внешние накопители (HDD, SSD, USB-флешки) требуют тщательной очистки, так как удаление файлов не стирает данные физически.

2.1 Методы очистки. Программная перезапись:

Стандарты:

- DoD 5220.22-M: 3 прохода (нули, единицы, случайные значения).
- Gutmann: 35 проходов для максимальной безопасности (устаревший, но используется для параноидальных сценариев).

Инструменты:

- DBAN (Darik's Boot and Nuke) для HDD.
- Parted Magic для SSD и HDD.

Шифрование с последующим удалением ключей. Суть: Данные шифруются, а ключ удаляется, делая информацию недоступной. Пример: Использование BitLocker (Windows) или LUKS (Linux) с последующим форматированием.

2. Физическое уничтожение:

a. Для HDD: Размагничивание (дегауссер), дробление дисков.

b. Для SSD: Измельчение чипов памяти, так как перезапись ненадежна из-за износоустойчивых алгоритмов.

Команда Secure Erase для SSD. Особенность: Сбрасывает ячейки памяти в исходное состояние, минуя контроллер.

Инструменты: Samsung Magician, HDAT2.

2.2 Особенности для разных типов накопителей:

a. HDD. Данные можно восстановить даже после форматирования. Требуется многократная перезапись.

b. SSD. Из-за технологии wear leveling и резервных блоков, стандартные методы перезаписи неэффективны. Лучше использовать Secure Erase или шифрование.

c. USB-флешки. Аналогичны HDD, но из-за низкого качества памяти часто выходят из строя после 3–5 циклов перезаписи.

Очистка оперативной памяти и внешних накопителей – критически важный этап для предотвращения утечек данных. Для ОЗУ эффективны методы принудительного обнуления и шифрование, а для накопителей – комбинация программной перезаписи, Secure Erase и физического уничтожения. Выбор метода зависит от типа носителя и уровня конфиденциальности данных. Всегда используйте проверенные стандарты (DoD 5220.22-M) и инструменты (DBAN, BitLocker), чтобы минимизировать риски восстановления информации.

Литература

1. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности. Ч. 4. Настройка подсистем СЗИ: учеб. пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» / В. В. Подтопельный. – Калининград: Изд-во БГАРФ. – 2021. – 97 с. – Библиогр.: с. 96–97. – ISBN 978-5-7481-0470-8 (6 авт. л.) (с. 29–94).

Контрольные вопросы

1. Приведите способы очистки (обнуление, обезличивание) освобождаемых областей оперативной памяти ЭВМ.
2. Приведите способы очистки (обнуление, обезличивание) внешних накопителей.
3. Почему простое удаление файлов недостаточно для очистки внешних накопителей?
4. Чем Secure Erase для SSD отличается от перезаписи?
5. Какие риски возникают при использовании Cold Boot атак?
6. Почему метод Гутмана считается избыточным для современных HDD?
7. Как шифрование помогает безопасно очистить данные?
8. Какие инструменты подходят для очистки оперативной памяти в реальном времени?

Тема 3.7 Системы активного аудита и АПКШ

Перечень изучаемых вопросов

1. Программно-аппаратные средства обеспечения информационной безопасности в вычислительных сетях.
2. Аппаратно-программные комплексы шифрования. Аудит ИБ АИС.

Методические указания к изучению

Рассмотрите программно-аппаратные средства обеспечения информационной безопасности в вычислительных сетях с использованием средств фильтрации и маршрутизации, аппаратно-программные комплексы шифрования.

1. Программно-аппаратные средства обеспечения информационной безопасности в вычислительных сетях

Программно-аппаратные средства сочетают специализированное оборудование и программное обеспечение для защиты данных и инфраструктуры. Они обеспечивают высокую производительность и надежность, критически важные для современных сетей.

Примеры средств:

1. Межсетевые экраны (фаерволы):

Аппаратные: Cisco ASA, FortiGate – обрабатывают трафик на уровне сети, минимизируя задержки.

Программные: pfSense, iptables – гибкие решения для малых сетей.

Функции: Фильтрация пакетов, блокировка DDoS-атак, контроль доступа.

2. Системы обнаружения и предотвращения вторжений (IDS/IPS):

Аппаратные решения: IBM QRadar, Palo Alto Networks – анализируют трафик в реальном времени.

Программные модули: Snort, Suricata – интегрируются в существующую инфраструктуру.

3. VPN-шлюзы:

Аппаратные: Juniper SRX, Check Point – обеспечивают шифрование трафика на уровне оборудования.

Программные: OpenVPN, WireGuard – подходят для облачных сред.

4. UTM-устройства (Unified Threat Management):

Примеры: Sophos UTM, WatchGuard – объединяют фаервол, антивирус, IDS/IPS, фильтрацию контента.

Преимущества аппаратной реализации:

- Высокая производительность при обработке большого объема данных.
- Защита от атак, направленных на программные уязвимости.
- Надежное хранение ключей и сертификатов в защищенных модулях (TPM, HSM).

2. Аппаратно-программные комплексы шифрования

Эти комплексы обеспечивают криптографическую защиту данных на всех этапах их жизненного цикла: при передаче, хранении и обработке.

Ключевые компоненты:

1. Аппаратные модули безопасности (HSM):

Назначение: Генерация, хранение и управление криптографическими ключами.

Примеры: Thales Luna HSM, YubiHSM – соответствуют стандартам FIPS 140-2, ГОСТ Р 34.10-2012.

Применение: Банковские транзакции, цифровые подписи, PKI.

2. Сетевые шифраторы:

Примеры: Cisco IPSec VPN, ViPNet – шифруют трафик между узлами сети.

Стандарты: AES-256, ГОСТ 28147-89.

3. Шифрование данных на носителях:

Аппаратные решения: Samsung SSD с AES-256, USB-токены с шифрованием.

Программные инструменты: VeraCrypt, BitLocker.

Особенности:

- Сертификация: Комплексы должны соответствовать требованиям регуляторов (ФСТЭК, ФСБ в России).
- Производительность: Аппаратные ускорители (например, Intel AES-NI) снижают нагрузку на CPU.

3. Аудит информационной безопасности автоматизированных информационных систем (АИС)

Аудит ИБ – систематическая проверка соответствия системы требованиям безопасности, выявление уязвимостей и оценка эффективности защитных мер.

Этапы аудита:

1. Планирование:

Определение целей, границ аудита, выбор стандартов (ISO 27001, PCI DSS).

2. Сбор информации:

Анализ политик безопасности, конфигураций сетевых устройств, журналов событий.

3. Тестирование:

Инструменты: Nessus (сканирование уязвимостей), Metasploit (пентест), Wireshark (анализ трафика).

Методы: Проверка на соответствие модели угроз, анализ рисков.

4. Отчетность:

Документирование выявленных уязвимостей (например, неправильно настроенные ACL).

Рекомендации по устранению проблем (обновление ПО, настройка MFA).

Автоматизация аудита:

- SIEM-системы: Splunk, IBM QRadar – агрегируют и анализируют логи в реальном времени.
- Средства анализа соответствия: Qualys, Rapid7 — проверяют настройки на соответствие стандартам.

Тренды:

- Использование ИИ: Для обнаружения аномалий и прогнозирования угроз.
- Аудит облачных сред: Проверка конфигураций AWS, Azure на соответствие Shared Responsibility Model.

Современные вычислительные сети требуют многоуровневой защиты, сочетающей программные и аппаратные решения. Аппаратно-программные

комплексы шифрования обеспечивают безопасность данных, а регулярный аудит ИБ помогает выявлять и устранять уязвимости. Ключевыми факторами успеха являются:

- Использование сертифицированных решений (HSM, сетевые шифраторы).
- Интеграция автоматизированных инструментов аудита (SIEM, сканеры уязвимостей).
- Следование международным и национальным стандартам безопасности.

Только комплексный подход позволяет противостоять эволюционирующим киберугрозам, включая атаки с использованием ИИ и целевые APT-кампании.

Литература

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный (с. 84–179).

Контрольные вопросы

1. Укажите специфику программно-аппаратных средствх обеспечения информационной безопасности в вычислительных сетях.
2. Укажите специфику аппаратно-программные комплексы шифрования.
3. Приведите специфику аудит ИБ АИС.
4. Чем аппаратный межсетевой экран отличается от программного?
5. Какие преимущества обеспечивают HSM при хранении криптографических ключей?
6. Назовите этапы проведения аудита информационной безопасности.
7. Почему UTM-устройства считаются комплексным решением для защиты сетей?
8. Какие стандарты шифрования актуальны для российских организаций?
9. Как SIEM-системы упрощают процесс аудита?

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ

Лабораторные занятия направлены на углубление знаний и закрепление основных понятий и методов, изучаемых в рамках дисциплины. Основная цель занятий – научиться применять теоретические знания для решения прикладных задач.

Рекомендации к подготовке

1. Изучение лекционного материала и конспектов:

– перед лабораторными занятиями необходимо проработать соответствующие разделы лекционного материала;

– рекомендуется вести конспект занятий и дополнительно пересмотреть его перед началом лабораторного занятия.

2. Проработка учебной литературы:

– используйте основные учебники и методические пособия, рекомендованные преподавателем;

– для глубокого понимания теории рекомендуется обращаться к дополнительной литературе.

3. Решение типовых задач:

– проработать примеры и типовые задачи из учебных материалов, методических указаний;

– выполните задачи, предложенные преподавателем для самостоятельной работы, что обеспечит лучшее понимание методов и подходов к решению задач.

4. Подготовка вопросов:

– составьте список вопросов по материалу, вызвавшему затруднения;

– обсуждение этих вопросов на лабораторном занятии поможет устранить пробелы в знаниях.

5. Вопросы для самоконтроля:

– перед каждым занятием рекомендуется проверять себя с помощью вопросов для самоконтроля из методических указаний к лекционным занятиям. Это позволит оценить уровень своей подготовки.

Тематический план лабораторных занятий приводится в разделе «Тематический план».

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ

В лекциях по предмету излагаются основные знания по курсу дисциплины. Самостоятельная работа имеет особое значение для прочного усвоения материала. Она помогает научиться правильно, ориентироваться в научной литературе, самостоятельно мыслить и находить правильные ответы на возникающие вопросы. В ходе всех видов занятий происходит углубление и закрепление знаний студентов, вырабатывается умение правильно излагать свои мысли.

Самостоятельная работа выполняет ряд функций, к которым относятся:

- развивающая (повышение культуры умственного труда, приобщение к творческим видам деятельности, обогащение интеллектуальных способностей студентов);
- информационно-обучающая (учебная деятельность студентов на аудиторных занятиях, неподкрепленная самостоятельной работой, становится малорезультативной);
- ориентирующая и стимулирующая (процессу обучения придается профессиональное ускорение);
- воспитывающая (формируются и развиваются профессиональные качества специалиста):
- исследовательская (новый уровень профессионально-творческого мышления).

В основе самостоятельной работы студентов лежат принципы: самостоятельности, развивающе-творческой направленности, целевого планирования, личностно-деятельностного подхода.

Самостоятельная работа студентов проводиться с целью:

– систематизации и закрепления полученных теоретических знаний и практических умений студентов;

– углубления и расширения теоретических знаний;

– формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;

– развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

– формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

– развития исследовательских умений.

Для достижения указанной цели студенты на основе плана самостоятельной работы должны решать следующие задачи:

– изучить рекомендуемые литературные источники:

– изучить основные понятия, представленные в глоссарии;

– ответить на контрольные вопросы:

– решить предложенные задачи, кейсы, ситуации;

– выполнить контрольные и курсовые работы.

Работа студентов в основном складывается из следующих элементов:

1. Изучение и усвоение в соответствии с учебным планом программного материала по всем учебным дисциплинам:
2. Выполнение письменных контрольных и курсовых работ;
3. Подготовка и сдача зачетов, курсовых работ, итоговых экзаменов:
4. Написание и защита дипломной работы.

Самостоятельная работа включает такие формы работы, как:

- индивидуальное занятие (домашние занятия) – важный элемент в работе студента по расширению и закреплению знаний;
- конспектирование лекций;
- получение консультаций для разъяснений по вопросам изучаемой дисциплины;
- подготовка ответов на вопросы тестов;
- подготовка к экзамену;
- выполнение контрольных, курсовых проектов и дипломных работ;
- подготовка научных докладов, рефератов, эссе;
- анализ деловых ситуаций (мини кейсов) и др.

Содержание внеаудиторной самостоятельной работы определяется в соответствии с рекомендуемыми видами заданий в соответствии с рабочей программой учебной дисциплины. Распределение объема времени на внеаудиторную самостоятельную работу в режиме дня студента не регламентируется расписанием.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференциальный характер, учитывать специфику специальности, изучаемой дисциплины, индивидуальные особенности студента.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

Для овладения знаниями:

- чтение текста (учебника, первоисточника, дополнительной литературы);
- составление плана текста;
- конспектирование текста;
- выписки из текста;
- работа со словарями и справочниками;
- исследовательская работа;
- использование аудио- и видеозаписи;
- работа с электронными информационными ресурсами и ресурсами Internet:

Для закрепления и систематизации знании:

- работа с конспектом лекции (обработка текста);
- повторная работа над учебным материалом (учебника, первоисточника,

дополнительной литературы, аудиовидеозаписей):

- составление плана и тезисов ответа;
- выполнение тестовых заданий;
- ответы на контрольные вопросы;
- аннотирование, реферирование, рецензирование текста;
- подготовка сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов;
- работа с компьютерными программами;
- подготовка к сдаче экзамена;

Для формирования умений:

- решение задач и упражнений по образцу;
- решение вариативных задач и упражнений:
- выполнение расчетно-графических работ;
- решение ситуационных производственных (профессиональных) задач;
- участие в научных и практических конференциях;
- проектирование и моделирование разных видов и компонентов профессиональной деятельности;
- создание проспектов, проектов, моделей;
- экспериментальная работа, участие в НИР;
- рефлексивный анализ профессиональных умений с использованием аудиовидеотехники и компьютерных расчетных программ, и электронных практикумов;
- подготовка курсовых проектов, работ и дипломных работ;

Правильная организация самостоятельных учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет привить студентам умения и навыки в овладении, изучении, усвоении и систематизации приобретаемых знаний в процессе обучения, обеспечивать высокий уровень успеваемости в период обучения, привить навыки повышения профессионального уровня в течение всей трудовой деятельности.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КУРСОВОМУ ПРОЕКТУ

Подробные указания приведены в учебно-методическом пособии по выполнению курсовых проектов для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» по дисциплине «Программно-аппаратные средства защиты информации»

Цели каждой отдельной курсового проекта должны раскрывать выбранную студентом тему. Курсовой проект предназначена для углубления студентами теоретических и практических навыков в области обеспечения информационной безопасности с помощью средств защиты информации. Современные требования к специалистам предполагают не только глубокое знание теоретических основ и принципов использования информационных технологий. Будущие специалисты должны иметь четкое представление обо всех этапах создания и эксплуатации информационных технологий, уметь осуществлять выбор из широкого арсенала современных средств и методов защиты информации в системах наиболее адекватные поставленной задаче. Курсовой проект – это одна из форм учебной (творческой и научно-исследовательской) работы, ее выполнение является обязательным для всех студентов очной и заочной форм обучения. Выполнение курсового проекта представляет собой самостоятельное решение студентом под руководством преподавателя частной задачи или проведение исследования по одному из вопросов, изучаемых в цикле специальных дисциплин (по ГОС ВПО) или в дисциплинах профессионального цикла (по ФГОС ВПО). Основной целью выполнения курсовых проектов является закрепление, углубление и обобщение знаний, полученных студентом за время теоретического и практического обучения, расширение объема профессионально значимых умений и навыков.

Различия курсовой работы и курсового проекта в следующем: курсовой проект в обязательной практической части предполагает работу над сложной расчетной и (или) проектной задачей, содержит описание хода решения задач и отчетные данные. Как правило, курсовой проект характерен для информационных или инженерных направлений, или специальностей.

Содержание курсовых проектов должно отвечать учебным задачам дисциплины, увязываться с последующей работой выпускников по специальности /направлению подготовки.

Поэтому в цели и задачи курсового проекта входят:

1) закрепление практических навыков настройки политик безопасности операционных систем, полученных на лабораторных занятиях по дисциплине «Безопасность операционных систем»;

2) углубление теоретических и практических знаний в области методологии отладки политик безопасности операционных систем;

3) развитие навыков самостоятельного планирования задач защиты операций и ключевой информации операционных систем;

4) получение опыта сбора регистрируемых событий, и обработки регистрируемых событий в операционной системе;

5) приобретение навыков создания резервных копий операционных систем.

Выполнение курсового проекта позволяет расширить и закрепить приобретенные студентом в ходе обучения в вузе теоретические знания и продемонстрировать полученные навыки по самостоятельной постановке и решению конкретной задачи, а также продемонстрировать владение профессиональными навыкам в области защиты информации.

При выполнении курсового проекта обучающимся рекомендуется использование элементов дистанционных образовательных технологий с использованием информационных и учебно-методических ресурсов. При этом график курсового проекта должен определяться количеством часов, указанным в учебном плане.

Важнейшими требованиями при выполнении курсового проекта для студента являются ее самостоятельность и актуальность, связанная с решением вопросов по заданиям или по тематике работ промышленных, коммерческих или научно-исследовательских организаций; использованием современной программной и аппаратной базы; справочных материалов; новейших методов организации расчетов, проектирования и исследований.

Обучающийся выбирает тему курсового проекта из числа предложенных тем. При выборе темы курсового проекта (КР) необходимо учесть возможность дальнейшего ее развития, углубления и конкретизации, а также использования в курсовой работе.

Обучающийся может предложить свою тему с обоснованием целесообразности ее разработки и при согласовании с заведующим кафедрой и/или научным руководителем.

Выбранная тема курсового проекта должна быть согласована с научным руководителем. Изменения темы курсового проекта могут быть внесены только после согласования с научным руководителем.

При выборе темы курсового проекта необходимо учитывать следующие условия:

– соответствие темы курсового проекта содержанию дисциплины, по которой выполняется курсовой проект, актуальность проблемы;

– наличие специальной литературы и возможность получения фактических данных, необходимых для анализа;

– собственные научные интересы и способности обучающегося; преемственность исследований, начатых в предыдущих курсовых проектах и в период учебных практик;

– исключение по возможности дублирования (дословного совпадения формулировок) тем курсовых проектов, выполняемых обучающимися (группой обучающихся).

Также при самостоятельном определении темы студенту требуется учесть свой опыт в выбранной сфере, наличие соответствующих знаний и навыков, а

также имеющихся наработок по предполагаемой тематике. Это, прежде всего, относится к тем, кто долго собирал и обрабатывал материал по той или иной проблематике, участвовал в НИРС, научных конференциях, имеет публикации в научных журналах, сборниках и т. д. Научный руководитель может быть преподаватель выпускающей кафедры

Студенту следует периодически информировать научного руководителя о ходе выполнения курсового проекта, консультироваться по вызывающим затруднения или сомнения теоретическим и практическим вопросам, обязательно ставить в известность о возможных проблемах в выполнении работы и её содержания. Изменение выбранной ранее темы курсового проекта возможно при согласовании с научным руководителем.

Курсовой проект выполняется студентом в период семестра, когда по учебному плану изучается соответствующая дисциплина.

Курсовой проект представляет собой решение практической, научно-исследовательской задачи одной из актуальных проблем в области защиты операционных систем,

Объектами курсового проекта могут быть методы поиска уязвимостей операционных систем, методы анализа уязвимости операционных, способы повышения защищенности операционных систем, специфика комплектования системного обеспечения в целях повышения информационной безопасности.

При выполнении курсового проекта должно быть предусмотрено:

– обоснование актуальности и важности решаемой задачи обеспечения информационной безопасности выбранного объекта;

– анализ проблемной области защиты операционных систем;

– определение, анализ возможных путей и способов исследования и описание выбранных методов и средств решения поставленных задач;

– методы и способы решения проблем безопасности операционных систем.

При определении темы и соответственно порядка разработки курсового проекта можно придерживаться следующего плана:

– точная формулировка темы, целей и задач выполнения курсового проекта;

– изучение специфики проблемной области;

– выявление уже существующих решений и определение их эффективности

– обоснование предложений по решению проблем в области информационной защиты операционных систем;

– реализация предложенных средств и методов защиты, исследования меры защищенности операционных систем и их компонентов;

– проверка работоспособности предложенных мер защиты.

Курсовой проект предусматривает **следующие этапы:**

1. **Подготовка** к выполнению курсового проекта заключается в изучении литературы по выбранной проблеме, сборе исходных данных по рассматриваемым проблемам. На этом этапе изучаются цели

функционирования и развития объекта, его обеспеченность средствами защиты, каналы уязвимости, Студент собирает, обобщает и систематизирует материалы, необходимые для разработки предложений Полученные материалы используются во введении и аналитической части работы.

2. Разработка темы. На основе собранных и обобщенных материалов, формулируются способы решения задач и разрабатываются алгоритмы решения задач, определяется специфика и порядок их реализации, реализуются предложенные решения, обосновывается эффективность разработки, исследований, решений.

3. Этап включает оформление курсового проекта. При этом выполняется:

– систематизация и обработка материалов курсового проекта;

– отбор материала для оформления содержательной части работы и составление структуры ее изложения, подготовка необходимого иллюстративного материала и т. д.;

– определение направлений и основного содержания предложений, выявление необходимости дополнительного сбора материалов; формирование чернового варианта разработки в целом;

– сбор дополнительных материалов, детальная разработка и обоснование выдвинутых предложений;

– уточнение аналитической и исследовательской части работы;

– редактирование и окончательное оформление отобранного материала;

– оформление иллюстративного материала.

4. Заключительным этапом подготовки курсового проекта к защите является предъявление ее преподавателю ИБ. К этому моменту курсовой проект должна быть подписана студентом.

Примерные темы курсовых проектов

1. Анализ методов и средств анализа защищенности беспроводных сетей.
2. Разработка OCR-приложения для распознавания утечки информации с ОС.
3. Обеспечение безопасности с использованием IOT и NFC-технологий.
4. Поиск и удаление вредоносных объектов из DOCX и PDF файлов».
5. Средства обеспечения информационной безопасности проводных сетей общего доступа, методология и анализ применяемых решений.
6. Средства обеспечения информационной безопасности банков данных.
7. Использование AVZ как антивирусной программы для борьбы вирусами.
8. Анализ критических характеристик линий связи с точки зрения обеспечения защиты информации.
9. Использование ЭЦП для обеспечения защиты информации при использовании системы электронного документооборота.
10. Обеспечение защиты конфиденциальной информации в распределённых системах разграничения доступа.

11. Установка, настройка и устранение неисправностей в процессе эксплуатации модуля InfoWatch Device Monitor.

12. Информационная система мониторинга и координации деятельности сотрудников информационно-технического отдела.

13. Инструментальные средства анализа рисков информационной безопасности.

14. Сравнительный и оценочный анализ международных стандартов в области информационной безопасности и управления рисками.

15. Оценочный анализ методов и средств тестирования системы защиты вычислительных сетей (аудита информационной безопасности).

16. Анализ рисков в области защиты информации.

17. Управление рисками и международные стандарты.

18. Технологии анализа рисков.

19. Инструментальные средства анализа рисков.

20. Аудит безопасности и анализ рисков.

21. Анализ защищенности информационной системы.

22. Обнаружение атак и управление рисками.

23. Оценка критичности сетевой атаки.

24. Сигнатуры как основной механизм выявления атак.

25. Анализ сетевого трафика и анализ контента.

26. IDS как средство управления рисками. Типовая и оптимальная архитектура системы выявления атак. Стандарты, определяющие правила взаимодействия между компонентами системы выявления атак.

27. CIDF. CVE – тезаурус уязвимостей.

6. ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Текущая аттестация

Преподаватель вправе выбрать методику оценивания знаний студентов: традиционная зачетно-экзаменационная либо балльно-рейтинговая**.** Выбрана традиционная зачетно-экзаменационная методика оценивания знаний. Предусматриваются: дифференцированный зачет, экзамен, курсовой проект.

В ходе изучения дисциплины студентам предстоит пройти следующие этапы текущей аттестации: контроль самостоятельной работы по темам дисциплины, контроль выполнения лабораторных работ.

К оценочным средствам текущего контроля успеваемости относятся:

– тестовые задания открытого и закрытого типов.

Промежуточная аттестация в форме зачета проходит по результатам прохождения всех видов текущего контроля успеваемости. В отдельных случаях (при не прохождении всех видов текущего контроля) зачет может быть проведен в виде тестирования.

Экзамен может проводиться как в традиционной форме, так и в виде экзаменационного тестирования. Тестовые задания для проведения экзаменационного тестирования приведены в фонде оценочных средств по дисциплине.

Критерии оценки результатов освоения дисциплины

Универсальная система оценивания результатов обучения включает в себя системы оценок: 1) «отлично», «хорошо», «удовлетворительно», «неудовлетворительно»; 2) «зачтено», «не зачтено»; 3) 100-балльную/процентную систему и правило перевода оценок в пятибалльную систему (таблица 2).

Таблица 2 – Система оценок и критерии выставления оценки

Система оценок / Критерий	**2**	**3**	**4**	**5**
	0–40 %	**41–60 %**	**61–80 %**	**81–100 %**
	«неудовлетво-рительно»	**«удовлетво-рительно»**	**«хорошо»**	**«отлично»**
	«не зачтено»	**«зачтено»**		
1 Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между	Обладает минималь-ным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект

Система оценок / Критерий	2	3	4	5
	0–40 %	**41–60 %**	**61–80 %**	**81–100 %**
	«неудовлетво-рительно»	**«удовлетво-рительно»**	**«хорошо»**	**«отлично»**
	«не зачтено»	**«зачтено»**		
	собой)			
2 Работа с информацией	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпрети-ровать и систематизи-ровать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
3 Научное осмысление изучаемого явления, процесса, объекта	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставлен ной информации	В состоянии осуществлять систематичес-кий и научно-корректный анализ предоставлен-ной информации, вовлекает в исследование новые релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
4 Освоение стандартных алгоритмов решения профессиональ ных задач	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

Оценивание тестовых заданий закрытого типа осуществляется по системе зачтено/не зачтено («зачтено» – 41–100 % правильных ответов; «не зачтено» – менее 40 % правильных ответов) или пятибалльной системе (оценка «неудовлетворительно» – менее 40 % правильных ответов; оценка «удовлетворительно» – от 41 до 60 % правильных ответов; оценка «хорошо» – от 61 до 80 % правильных ответов; оценка «отлично» – от 81 до 100 % правильных ответов).

Тестовые задания открытого типа оцениваются по системе «зачтено/не зачтено». Оценивается верность ответа по существу вопроса, при этом не учитывается порядок слов в словосочетании, верность окончаний, падежи.

Завершающим этапом изучения дисциплины является итоговая аттестация, представляющая собой экзамен.

Допуск к итоговой аттестации возможен при:

– наличии всех выполненных, сданных (проверенных, защищенных) лабораторных работах, наличии отчётов по ним;

– наличии показателей приемлемого уровня освоения материалов курса: более 50 % посещений от общего числа требуемых по учебному плану.

Примерные вопросы к зачету/экзамену по дисциплине

Вопросы к зачету

1. Надежность информации.
2. Интегральная информационная безопасность.
3. Основные этапы жизненного цикла информации.
4. Элементы информационной базы АСОД.
5. Уязвимость информации.
6. Типовые структурные компоненты АСОД.
7. Типы дестабилизирующих факторов.
8. Причины нарушения целостности информации.
9. Каналы несанкционированного получения информации без доступа нарушителя к элементам ЭВТ, АСОД.
10. Каналы несанкционированного получения информации с доступом нарушителя к элементам АСОД, но без их изменений.
11. Каналы несанкционированного получения информации с доступом нарушителя к элементам АСОД с их изменением.
12. Классификация угроз безопасности.
13. Основные методы защиты информации в вычислительных системах.
14. Общая схема идентификации и установления подлинности пользователя.
15. Метод проверки подлинности на основе простого пароля.
16. Метод проверки подлинности на основе динамически изменяющегося пароля.
17. Организация контроля информационной целостности.
18. Задачи, решаемые аппаратными средствами защиты.
19. Классификация аппаратных средств защиты.

20. Классификация программных средств защиты.
21. Виды программных средств защиты.
22. Средства защиты данных.
23. Средства защиты от копирования.
24. Средства защиты информации о разрушения.
25. Концепция диспетчера доступа.

Вопросы к экзамену

1. Методы управления безопасностью сетей.
2. Основные требования защиты сетей и возможные им угрозы.
3. Цели и задачи защиты информации в вычислительных сетях.
4. Перечень и содержание сервисов безопасности.
5. Стандарты сервисов безопасности.
6. Классификация видов услуг механизмов защиты.
7. Сущность методов распределения ключей при использовании механизмов цифровой подписи данных, передаваемых в сетях.
8. Основные положения концепции защиты информации в эталонной модели взаимодействия открытых сетей.
9. Назначение, задачи системы защиты СЗИ AURA.
10. Общее содержание функций подсистемы идентификации и аутентификации СЗИ AURA.
11. Общее содержание функций подсистемы разграничения доступа к ресурсам СЗИ AURA.
12. Общее содержание функций подсистемы контроля целостности СЗИ AURA.
13. Общее содержание функций подсистемы регистрации событий СЗИ AURA.
14. Общее содержание функций подсистемы управления средствами защиты (администрирования) СЗИ AURA.
15. Назначение, задачи, классификация межсетевых экранов.
16. Назначение, задачи прокси-серверов.
17. Характеристика систем активного аудита.
18. Технологии и средства защиты процессов переработки информации в Интернете.
19. Основное содержание информационной безопасности в Интранете.
20. Назначение, состав и возможности системы защиты информации Dallas Lock.
21. Назначение, состав и возможности программно-аппаратного комплекса защиты информации Аккорд.
22. Назначение, состав и возможности программно-аппаратного комплекса защиты информации Соболь – PCI.
23. Назначение, состав и возможности программно-аппаратного комплекса защиты информации Страж NT.
24. Назначение, состав и возможности системы защиты Secret Disk.

25. Методы и средства нейтрализации угроз.

26. Основные нормативные правовые документы по информатизации и защите информации.

27. Основные специальные меры по технической защите информации, обрабатываемой средствами вычислительной техники. (Требования ФСТЭК).

28. Основные принципы защиты от НСД.

29. Основные способы и направления обеспечения защиты от НСД.

30. Основная структура и содержание монитора обращений.

31. Основные модели нарушителей в автоматизированных системах.

32. Порядок обеспечения защиты от НСД к ПК при его оставлении без завершения сеанса работы.

33. Классификация вирусов и методов защиты от них.

34. Классы и виды антивирусных программ.

35. Методы выявления программ-шпионов.

36. Укажите стандарты (ГОСТ Р) и РД, применяемые при эксплуатации СрЗИ.

ЗАКЛЮЧЕНИЕ

В лекциях по предмету излагаются основные знания по курсу дисциплины. Самостоятельная работа имеет особое значение для прочного усвоения материала. Она помогает научиться правильно ориентироваться в научной литературе, самостоятельно мыслить и находить правильные ответы на возникающие вопросы. В ходе всех видов занятий происходят углубление и закрепление знаний студентов, вырабатывается умение правильно излагать свои мысли.

Правильная организация учебных занятий, их систематичность, целесообразное планирование рабочего времени позволяет привить студентам умения и навыки в овладении, изучении, усвоении и систематизации приобретаемых знаний в процессе обучения, обеспечивать высокий уровень успеваемости в период обучения, привить навыки повышения профессионального уровня в течение всей трудовой деятельности.

ЛИТЕРАТУРА

Основные источники

1. Маршаков, Д. В. Программно-аппаратные средства защиты информации: учеб. пособие / Д. В. Маршаков, Д. В. Фатхи. – Ростов-на-Дону: Донской ГТУ, 2021. – 228 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/237770 (дата обращения: 05.12.2024). – ISBN 978-5-7890-1878-1. – Текст : электронный.

2. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб. пособие по дисциплине «Программно-аппарат. средства защиты информации» для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем»: в 2 ч. / В. В. Подтопельный; Калинингр. гос. техн. ун-т. – Калининград: КГТУ, 2024. – ISBN 978-5-94826-691-6. – Текст : непосредственный. Ч. 1. Поиск и удаление вредоносных объектов в информационных системах. – 2024. – 121, [1] с. – ISBN 978-5-94826-692-3 (в обл.).

3. Жестовский, А. Г. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов высш. учеб. заведений, обучающихся по направлению «Информ. безопасность», по прогр. подгот. бакалавров, магистров, специалистов / А. Г. Жестовский, В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2018. – Текст : непосредственный. Ч. 2. Настройка систем защиты информации от несанкционированного доступа. – 2018. – 100 с. – ISBN 978-5-7481-0389-3.

4. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 3. Поиск и извлечение вредоносных программ в программной среде. – 2020. – 99 с.

5. Подтопельный, В. В. Программно-аппаратные средства обеспечения информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный; Федер. агентство по рыболовству [и др.]. – Калининград: БГАРФ, 2020. – Текст : непосредственный. Ч. 4. Настройка подсистем СЗИ. – 2021. – 97 с.

6. Подтопельный, В. В. Аудит информационной безопасности: учеб. пособие для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» всех форм обучения / В. В. Подтопельный; Федер. агентство по рыболовству [и. др.]. – Калининград: БГАРФ, 2023. – Текст : непосредственный. – Ч. 1. – 171 с. – ISBN 978-5-7481-0514-9.

Дополнительные источники

7. Национальная безопасность: учебник / В. И. Абрамов, М. А. Газимагомедов, К. К. Гасанов [и др.]; под ред. К. К. Гасанова, Н. Д. Эриашвили, О. А. Мироновой. – 3-е изд., перераб. и доп. – Москва: Юнити-Дана, 2023. – 288 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=700171 (дата обращения: 05.11.2024). – ISBN 978-5-238-03639-7. – Текст : электронный.

8. Программно-аппаратные средства защиты информации: учеб. пособие / С. А. Зырянов, М. А. Кувшинов, И. А. Огнев, И. В. Никрошкин. – Новосибирск: НГТУ, 2023. – 80 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/404549 (дата обращения: 05.12.2024). – ISBN 978-5-7782-4905-9. – Текст : электронный.

9. Жмуров, Д. Б. Программно-аппаратные средства защиты информации: учеб. пособие / Д. Б. Жмуров, С. В. Жуков. – Самара: Самарский университет, 2022. – 80 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URLhttps://e.lanbook.com/book/336515 (дата обращения: 05.12.2024). – ISBN 978-5-7883-1799-1. – Текст : электронный.

10. Бутин, А. А. Программно-аппаратные средства защиты информации: учеб. пособие / А. А. Бутин, Н. И. Глухов, С. И. Носков. – 2-е изд., перераб. и доп. – Иркутск: ИрГУПС, 2022. – 92 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/342113 (дата обращения: 05.12.2024). – Текст : электронный.

11. Тумбинская, М. В. Комплексное обеспечение информационной безопасности на предприятии: учебник для вузов / М. В. Тумбинская, М. В. Петровский. – 3-е изд., стер. – Санкт-Петербург: Лань, 2025. – 344 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/445253 (дата обращения: 05.12. 2024). – ISBN 978-5-507-52270-5. – Текст : электронный.

12. Программно-аппаратные средства обеспечения информационной безопасности: лаб. практикум для студентов специальности 10.05.03 «Информ. безопасность автоматизир. систем» / Федер. агентство по рыболовству, Калининград. гос. техн. ун-т, Балт. гос. акад. рыбопромыслового флота; сост.: А. Г. Жестовский, В. В. Подтопельный. – 2-е изд., перераб. и доп. – Калининград: БГАРФ, 2019. – Режим доступа: для авториз. пользователей. – URL: https://lib.klgtu.ru/web/index.php (дата обращения: 05.11.2024). – ISBN 978-5-238-03639-7. – Текст : электронный. Ч. 1. Защита компьютерной информации и компьютерных систем от вредоносных программ.

Учебно-методические пособия, нормативная литература

13. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб.-метод. пособие по изучению дисциплины для студентов специальности 10.05.03 «Информационная безопасность автоматизированных

систем». – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 42 с. – URL: https://klgtu.ru/vikon/sveden/files/kek/UMP_Programmnoapparatnye_sredstva_zaschity_informacii(1).pdf (дата обращения: 01.11.2024). – Текст : электронный.

14. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб.-метод. пособие по выполнению курсовых проектов для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем». – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 56 с. – URL: https://klgtu.ru/vikon/sveden/files/eiq/UMP_Programmnoapparatnye_sredstva_zaschity_informacii_(kursovoi_proekt).pdf (дата обращения: 01.11.2024). – Текст : электронный.

15. Подтопельный, В. В. Программно-аппаратные средства защиты информации: учеб.-методич. пособие по выполнению лабораторных работ по дисциплине для студентов специальности 10.05.03 Информационная безопасность автоматизированных систем. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 186 с. – Режим доступа: для авториз. пользователей. – URL: https://eios.klgtu.ru/course/view.php?id=9328 (дата обращения: 08.12.2024). – Текст : электронный.

16. Программно-аппаратные средства защиты информации: учеб.-метод. пособие / С. И. Штеренберг, А. М. Гельфанд, Д. В. Рыжаков, Р. А. Фатхутдинов. – Санкт-Петербург: СПбГУТ им. М. А. Бонч-Бруевича, 2017. – 98 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/180093 (дата обращения: 08.12.2024). – Текст : электронный.

17. Булычёв, Г. Г. Программно-аппаратные средства защиты информации: учеб.-метод. пособие / Г. Г. Булычёв. – Москва: РТУ МИРЭА, 2022. – Ч. 1. – 2022. – 203 с. – Режим доступа: для авториз. пользователей. — Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/310781 (дата обращения: 08.12.2024). – ISBN 978-5-7339-1652-1. – Текст : электронный.

18. Булычёв, Г. Г. Программно-аппаратные средства защиты информации: учеб.-метод. пособие / Г. Г. Булычёв. – Москва: РТУ МИРЭА, 2022. – Ч. 2. – 2022. – 177 с. – Режим доступа: для авториз. пользователей. – Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/310784 (дата обращения: 08.12.2024). – ISBN 978-5-7339-1653-8. – Текст : электронный.

19. «Доктрина информационной безопасности Российской Федерации» (утв. Указом Президентом РФ 05.12.2016 № 646 (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

20. «Конституция Российской Федерации» (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020) (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

21. Федеральный закон от 28.12.2010 N 390-ФЗ «О безопасности» (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

22. Федеральный закон от 27.07.2006 N 149-ФЗ «Об информации, информационных технологиях и о защите информации» (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

23. Федеральный закон от 27.07.2006 N 152-ФЗ «О персональных данных» (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

24. Закон РФ от 21.07.1993 N 5485-1 «О государственной тайне» (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

25. Указ Президента РФ от 06.03.1997 N 188 «Об утверждении Перечня сведений конфиденциального характера» (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

26. «ГОСТ Р 50739-95. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования» (принят и введен в действие Постановлением Госстандарта РФ от 09.02.1995 N 49) (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

27. «ГОСТ Р 50922-2006. Национальный стандарт Российской Федерации. Защита информации. Основные термины и определения» (утв. и введен в действие Приказом Ростехрегулирования от 27.12.2006 N 373-ст) в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

28. «Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации» (утв. Решением Гостехкомиссии России от 30.03.1992) (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

29. «Руководящий документ. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации» (утв. Решением Гостехкомиссии России 30.03.1992) (в действующей редакции). – Режим доступа: для авториз. пользователей из справ.-правовой системы КонсультантПлюс. – Текст: электронный.

Локальный электронный методический материал

Владислав Владимирович Подтопельный

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 7,1. Печ. л. 5,7.

Издательство федерального государственного бюджетного образовательного учреждения высшего образования «Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1