МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования «Южный федеральный университет»
(ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ)

УТВЕРЖДАЮ

Руководитель образовательной программы

_____ / А.Н. Попов /

«____» _____ 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Безопасность информационных технологий

Ростов-на-Дону – Таганрог 2020

Составитель рабочей программы:

_____ Брюхомицкий Ю.А.

_____ Поликарпов С.В.

«____» _____ 2020 г.

Программа одобрена на заседании кафедры Синергетики и процессов управления

«____» _____ 2020 г., протокол № _____

Заведующий кафедрой:

_____ / Попов А.Н. /

«____» _____ 2020 г.

СОДЕРЖАНИЕ

I. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины «Безопасность информационных технологий» состоит в формировании у студентов базовых знаний по проблемам безопасности информационных технологий, методам и средствам обеспечения информационной безопасности, а так же получение студентами первичных навыков работы со средствами защиты информации.

Задачи изучения дисциплины. Задачи изучения дисциплины заключаются в том, чтобы дать основы знаний о:

- возможных угрозах безопасности данных и уязвимостях информационных систем;
- классификации методов и средств защиты информации;
- стандартизации в области информационной безопасности;
- методах и средствах криптографии и способах их применения;
- методах и средствах аутентификации субъектов, объектов и процессов;
- методах и механизмах авторизации субъектов в информационных системах;
- методах и средствах обеспечения безопасности операционных систем;
- средствах обнаружения и нейтрализации вредоносного ПО;
- методах и средствах защиты информации в компьютерных сетях;
- технических средствах защиты информации;
- организационно-правовом обеспечении информационной безопасности.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина относится к модулю обязательных профессиональных дисциплин образовательной программы.

Для изучения данной дисциплины необходимы знания, умения и навыки, формируемые предшествующими элементами образовательной программы:

Наименование дисциплины (модуля), практики	Требуемые знания, умения, навыки
Дискретная математика	Знания: - двоичной системы счисления и булевой алгебры; - комбинаторики; - теории вероятностей и математической статистики; - математической логики; - алгебры и теории чисел; - вычислительной математики; - дискретной математики.
Алгоритмизация и программирование	Знания: - языка программирования Си или Python; - структуры данных и алгоритмов обработки данных. Навыки: - работы в текстовом редакторе с подсветкой синтаксиса или в среде разработке для редактирования исходного кода программ на языке

Наименование дисциплины (модуля), практики	Требуемые знания, умения, навыки
	программирования Си или Python; - компилирования исходного кода программ (написанных на языке программирования Си) в исполняемые двоичные файлы; - отладки программ отладчиками.
Операционные системы	Знания: - архитектур операционных систем семейства Windows и Linux; - принципов распределения и разграничения ресурсов в многозадачных многопользовательских операционных системах. Навыки: - установки, запуска и настройки операционных систем семейства Windows и Linux, в том числе сетевых сервисов; - установки, настройки и запуска программного обеспечения в операционных системах Windows и Linux.

Знания, умения и навыки, формируемые данной дисциплиной, потребуются при освоении следующих элементов образовательной программы:

- преддипломная практика;
- подготовка выпускной квалификационной работы.

III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование следующих компетенций в соответствии с образовательным стандартом и образовательной программой:

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы достижения компетенции	Результаты обучения
ОПК-2 Способен решать задачи профессиональной деятельности с применением современных информационно-коммуникационных технологий, технических и программных средств, в том числе отечественного производства, и с учетом основных требований информационной безопасности и профессиональной этики.	**ОПК-2.1** Выбирает современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности.	**Знать:** - возможные угрозы безопасности данных и уязвимости информационных систем; - классификацию методов и средств защиты информации; - стандартизацию в области информационной безопасности.
	ОПК-2.3 Решает стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий	**Знать:** - организационно-правовое обеспечении информационной безопасности. **Владеть:** - навыками выработки основных требований к обеспечению информационной безопасности с использованием современных информационно-коммуникационных технологий.
	ОПК-2.4 Учитывает основные требования информационной безопасности при решении стандартных задачи профессиональной деятельности	**Знать:** - методы и средства криптографии и способов их применения; - методы и средства аутентификации субъектов, объектов и процессов; - методы и механизмы авторизации субъектов в информационных системах; - методы и средства обеспечения безопасности операционных систем; - средства обнаружения и нейтрализации вредоносного ПО; методы и средства защиты информации в

		компьютерных сетях; - технические средства защиты информации. **Уметь:** - решать задачи обеспечения информационной безопасности с использованием современных информационно-коммуникационных технологий.
(09.03.01) ОПК-5. Способен инсталлировать, администрировать и осуществлять наладку программного и аппаратного обеспечения информационных и автоматизированных систем **(02.03.03, 09.03.04) ОПК-5.** Способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства.	**ОПК-5.1** Инсталлирует и администрирует программное обеспечение на основе современных стандартов информационного взаимодействия систем	**Знать:** - методы и средства обеспечения безопасности операционных систем; - средства обнаружения и нейтрализации вредоносного ПО. **Владеть:** - навыками установки и настройки программного межсетевого экрана.
	ОПК-5.3 Администрирует и осуществляет наладку аппаратного обеспечения информационных и автоматизированных систем	**Знать:** - технические средства защиты информации. **Владеть:** - навыками работы с технические средства защиты информации.

IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 5 зачетных единицы, 180 часов
Форма отчетности: дифференцированный зачет

IV.1 Содержание дисциплины, структурированное по темам

№ п/п	Темы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Наименования оценочных средств
			Контактная работа			Самостоятельная работа	
			Лекции	Практические занятия	Лабораторные занятия		
	Модуль 1						
1	Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Критерии и стандарты защиты данных	3	4	–	–	10	Контрольная работа №1.
2	Методы и средства аутентификации. Контроль доступа. Политики и модели безопасности	3	4	–	–	10	Контрольная работа №1.
3	Техническая защита информации	3	6	–	18	36	Контрольная работа №1. Защита лабораторных работ №1- №4.
4	Организационно-правовое обеспечение информационной безопасности	3	4	–	–	10	Контрольная работа №1.
	Модуль 2						
5	Криптографические методы защиты информации и криптоанализ	3	6	–	8	12	Контрольная работа №2. Защита лабораторных работ №5-6.

№ п/п	Темы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Наименования оценочных средств
			Контактная работа			Самостоятельная работа	
			Лекции	Практические занятия	Лабораторные занятия		
6	Вредоносное программное обеспечение	3	4	–	4	10	Контрольная работа №2. Защита лабораторной работы №7.
7	Безопасность операционных систем	3	4	–	–	10	Контрольная работа №2.
8	Защита информации в компьютерных сетях	3	4	–	6	10	Контрольная работа №2. Защита лабораторной работы №8.
Итого часов		**3**	**36**	**–**	**36**	**108**	**Дифференцированный зачёт**

IV.2 План внеаудиторной самостоятельной работы

№ п/п	Темы дисциплины	Семестр	Вид самостоятельной работы	Сроки выполнения (нед.)	Затраты времени (часы)	Учебно-методическое обеспечение
	Модуль 1					
1	Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Критерии и стандарты защиты данных	3	- проработка и повторение материала лекций и литературы.	1-2	10	Конспект лекций. ОЛ: [1, 2]
2	Методы и средства аутентификации. Контроль доступа. Политики и модели безопасности	3	- проработка и повторение материала лекций и литературы.	3-4	10	ОЛ: [1-3]

№ п/п	Темы дисциплины	Семестр	Вид самостоятельной работы	Сроки выполнения (нед.)	Затраты времени (часы)	Учебно-методическое обеспечение
3	Техническая защита информации	3	- проработка и повторение материала лекций и литературы; - подготовка к лабораторной работе; - подготовка отчётов о выполнении лабораторной работы; - подготовка к защите отчётов о выполнении лабораторных работ.	5-7	36	ОЛ: [4] Описание лабораторных работ
4	Организационно-правовое обеспечение информационной безопасности	3	- проработка и повторение материала лекций и литературы.	8	10	ОЛ: [1-3]
	Модуль 2					
5	Криптографические методы защиты информации и криптоанализ	3	- проработка и повторение материала лекций и литературы; - подготовка к лабораторной работе; - подготовка отчётов о выполнении лабораторной работы; - подготовка к защите отчётов о выполнении лабораторных работ.	9-11	12	ОЛ: [5-7] ДЛ: [9] Описание лабораторных работ
6	Вредоносное программное обеспечение	3	- проработка и повторение материала лекций и литературы; - подготовка к лабораторной работе; - подготовка отчётов о выполнении лабораторной работы; - подготовка к защите отчётов о выполнении лабораторных работ.	12-13	10	ОЛ: [6, 7] Описание лабораторных работ
7	Безопасность операционных систем	3	- проработка и повторение материала лекций и литературы.	14-15	10	ОЛ: [6, 7] ДЛ: [8]

№ п/п	Темы дисциплины	Семестр	Вид самостоятельной работы	Сроки выполнения (нед.)	Затраты времени (часы)	Учебно-методическое обеспечение
8	Защита информации в компьютерных сетях	3	- проработка и повторение материала лекций и литературы; - подготовка к лабораторной работе; - подготовка отчётов о выполнении лабораторной работы; - подготовка к защите отчётов о выполнении лабораторных работ.	16-17	10	ОЛ: [6, 7] Описание лабораторных работ
Общая трудоемкость самостоятельной работы по дисциплине					**108**	–

IV.3. Содержание учебного материала

Модуль 1.

Раздел 1. Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Критерии и стандарты защиты данных

Тема 1. Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Информационная безопасность. Защита информации. Основные составляющие информационной безопасности**.** Компьютерная безопасность. Субъектно-объектная модель. Угрозы и уязвимости ИС. Классификация угроз безопасности ИС. Методы оценивания угроз безопасности КС. Классификация злоумышленников. Модель нарушителя. Классификация каналов проникновения в систему и утечки информации. Классификация методов и средств защиты информации.

Тема 2. Критерии и стандарты защиты данных. Критерии и стандарты защиты данных. Базовые понятия и принципы Общих Критериев. Функциональные требования безопасности. Требования доверия безопасности. Другие отечественные стандарты.

Раздел 2. Методы и средства аутентификации. Контроль доступа. Политики и модели безопасности

Тема 3. Методы и средства аутентификации. Основные понятия в технологиях идентификации и аутентификации. Парольная аутентификация. Аутентификация пользователей на основе модели «рукопожатия». Персональные средства аутентификации. Биометрические средства аутентификации. Аутентификация по информации, ассоциированной с субъектом.

Тема 4. Контроль доступа. Политики и модели безопасности. Субъектно-объектная модель. Монитор безопасности. Политики безопасности. Модели безопасности. Дискреционные модели безопасности. Мандатные модели безопасности. Ролевые модели безопасности.

Раздел 3. Техническая защита информации

Тема 5. Технические каналы утечки информации. Классификация и характеристика технических каналов утечки информации. Классификация и характеристика технических каналов утечки акустической (речевой) информации. Технические каналы утечки информации при ее передаче по каналам связи. Скрытое видеонаблюдение и съемки. Средства фоторазведки и фотодокументирования.

Тема 6. Портативные средства технической разведки. Портативные средства акустической разведки. Портативные средства радио-, радиотехнической разведки. Средства компьютерного шпионажа. Системы слежения за транспортными средствами. Автономные портативные технические средства разведки.

Тема 7. Методы и средства защиты от технической разведки. Поиск радиозакладок. Способы и аппаратура защиты телефонных линий. Средства защиты от НСД к акустической информации.

Раздел 4. Организационно-правовое обеспечение информационной безопасности.

Тема 8. Организационно-правовое обеспечение информационной безопасности. Основные определения в области информационного права. Классификация и виды информационных ресурсов. Информация ограниченного доступа. Государственная тайна и ее защита. Защита коммерческой тайны. Правовая защита профессиональной и служебной тайны. Правовая основа системы лицензирования и сертификации. Лицензирование деятельности по

защите ГТ. Сертификации средств защиты информации. Лицензирование и сертификация в области защиты конфиденциальной информации. Борьба с киберпреступностью.

Модуль 2.

Раздел 5. Криптографические методы защиты информации и криптоанализ.

Тема 9. Симметричные криптографические системы. Стойкость криптографических систем и алгоритмов. Основные методы криптоанализа шифров. Простейшие шифры и их основные свойства. Основные классы симметричных криптографических систем. Поточные шифры: основные требования, режимы функционирования. Блочные шифры: основные требования к стойкости и режимы функционирования. Основные структуры блочных шифров: SP-сети и сети Фейстеля. Российский стандарт блочного шифрования «Кузнечик»: основные параметры, структура раунда шифрования и принцип действия, процедура разворачивания исходного ключа в раундовые (рабочие) подключи, режимы использования. Стойкость, обеспечиваемая современными шифрами. Причины ненадёжности криптографических систем.

Тема 10. Функции хеширования: назначение и основные требования. Функции хеширования SHA-3 и GOST R 34.11-2012 (Streebog): основные параметры, структура и принцип действия преобразующей функции. Методы строгой аутентификации. Стандарт X.509. Протоколы аутентификации с симметричными алгоритмами шифрования. Протокол аутентификации и распределения ключей Нидхэма-Шредера.

Тема 11. Асимметричные криптографические системы: основные требования, односторонние функции и функции-ловушки. Асимметричные криптографические системы Эль Гамаля и Ривеста-Шамира-Адлемана (RSA). Асимметричная криптографическая система, основанная на проблеме Диффи-Хеллмана. Электронные цифровые подписи: основные требования, алгоритмы электронной цифровой подписи. Цифровые подписи, основанные на асимметричных криптографических алгоритмах. Цифровые сертификаты. Использование сертификатов для управления криптографическими ключами. Строгая аутентификация, основанная на асимметричных алгоритмах шифрования.

Раздел 6. Вредоносное программное обеспечение.

Тема 12. Классификация вредоносных программ. Жизненный цикл вредоносных программ. Основные каналы распространения вредоносных программ. Уязвимости программного обеспечения («переполнение буфера», «манипуляция указателя» и др.). База данных уязвимостей CVE.

Тема 13. Специализированные средства и методы выявления вредоносных программ: сигнатурное и эвристическое сканирование, контроль целостности, мониторинг информационных потоков, изолированная программная среда, программные ловушки. Антивирусные программы и комплексы.

Раздел 7. Безопасность операционных систем.

Тема 14. Основы безопасности операционных систем. Классификация угроз безопасности ОС, типичные атаки на ОС. Понятие защищённой ОС: основные определения, подходы к построению защищённых ОС, административные меры защиты, адекватная политика безопасности. Основные виды уязвимостей в ОС. Основные задачи аппаратного обеспечения защиты информации: управление оперативной памятью, планирование задач, синхронизация параллельных задач, обеспечение корректности совместного доступа к объектам, предотвращение тупиковых ситуаций. Аппаратная защита в процессорах с архитектурами x86 и x86-64: адресация

оперативной памяти, уровни привилегированности, защита сегментов оперативной памяти. Расширения команд процессоров AES-NI и NX-Bit (ND-Bit). Модуль доверенной платформы Trusted Platform Module (TPM). Аудит событий.

Тема 15. Разграничение доступа к ресурсам ОС. Системы контейнерной и полной изоляции (виртуализации) операционных систем и программного обеспечения (Docker, XEN, Qemu, VMWare и др.). Система принудительного контроля доступа SELinux. Технология рандомизации расположения адресного пространства ASLR. Средства шифрования в ОС Windows. Средства шифрования в ОС Linux. Архивирование файлов с применением шифрования. Программные и программно-аппаратные средства шифрования дисков.

Раздел 8. Защита информации в компьютерных сетях.

Тема 16. Технологии межсетевых экранов. Функции межсетевых экранов, особенности функционирования межсетевых экранов на различных уровнях модели OSI. Критерии оценки качества межсетевых экранов: общие требования; основные классы защищенности межсетевых экранов в соответствии с руководящими документами ФСТЭК России. Обзор современных межсетевых экранов. Примеры работы межсетевых экранов для различных сценариев передачи информации.

Тема 17. Технологии виртуальных частных сетей VPN. Классификация VPN. Протокол защиты сетевого уровня IPSe. Протоколы формирования защищенных каналов на сеансовом уровне SSL/TLS. Протоколы формирования защищенных каналов на канальном уровне L2F и L2TP. Примеры настройки и работы защищённых туннелей.

Тема 18. Технологии анализа защищенности. Средства анализа защищенности сетевых протоколов и сервисов. Технологии обнаружения сетевых атак: методы анализа сетевой информации, классификация систем обнаружения атак, компоненты и архитектура системы обнаружения атак. Особенности систем обнаружения атак на сетевом и операционном уровнях. Методы реагирования на атаки. Обзор современных средств обнаружения атак.

Перечень лабораторных работ

№ п/п	Название лабораторной работы	Количество часов
1	Исследование каналов утечки информации за счет побочных электромагнитных излучений.	4
2	Методы съёма информации по виброакустическому каналу и меры противодействия утечке информации по виброакустическому каналу.	4
3	Скремблер речевого сигнала.	4
4	Поиск неоднородностей в кабельных линиях.	6
5	Восстановление и формирование паролей пользователей.	4
6	Изучение принципов работы симметричного криптоалгоритма «Кузнечик».	4
7	Исследование уязвимости «переполнение буфера».	4
8	Настройка VPN-соединения между МСЭ PFSENSE (OpenVPN-PSK)».	6
Всего часов		**36**

V. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

По дисциплине предусмотрены следующие методы обучения и интерактивные формы проведения занятий:

- объяснительно-иллюстративные;
- контекстные (решение профессионально-ориентированных задач, разбор конкретных ситуаций)
- групповой работы (организация работы малых групп при выполнении лабораторных работ).

Для изучения дисциплины предлагается сочетание традиционных образовательных технологий в форме лекций с интерактивными элементами, информационными технологиями при выполнении лабораторных работ и проведении контрольных мероприятий.

Лекционные занятия проводятся в форме электронной презентации материалов дисциплины, которая предварительно распространяется среди студентов. Электронная презентация включает в себя основные положения и базовые понятия изучаемой дисциплины. В результате перед каждой лекцией студенты имеют возможность, в рамках самостоятельной работы, получить основные представления по теме лекции. На самой лекции основные положения и базовые понятия дополняются необходимыми пояснениями, деталями, примерами и обсуждением наиболее актуальных и проблемных вопросов. Это позволяет более продуктивно использовать время лекции и существенно увеличить объем учебного материала дисциплины. Изложение лекций может дополняться также интерактивными формами получения знаний: обсуждением конкретных вопросов, примеров, ситуаций.

В итоге студенты, помимо базовой части лекционного материала, изложенного лектором, приобретают индивидуальную вариативную составляющую знаний, отражающую нюансы своего личностного восприятия дисциплины.

Лабораторные работы охватывают основные разделы дисциплины и проводятся с использованием разнообразных программных средств (учебных пакетов программ, интерактивных платформ, программных эмуляторов) и аппаратно-программных средств (персональных средств аутентификации, средств контроля доступа, средств контроля каналов утечки информации). Защита лабораторных работ организуется в интерактивной форме с использованием практических примеров и ситуаций в сфере обеспечения безопасности информации и объектов информатизации.

Наряду с традиционными образовательными технологиями, для реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологий (ЭО и ДОТ) в ЭИОС Университета, включая систему электронного обучения ИКТИБ (lms.sfedu.ru). Лекционные аудиторные занятия и другие формы контактной работы обучающихся с преподавателем могут проводиться с использованием платформ Microsoft Teams, Cisco, Skype, Google Classroom, Zoom и др., что позволяет обеспечить онлайн и офлайн взаимодействие преподавателя с обучающимися в рамках дисциплины

Основными методами текущего контроля являются электронный учёт и контроль учебных достижений студентов (использование средств сервиса балльно-рейтинговой системы; ведение электронного журнала успеваемости и применение других средств контроля с использованием системы электронного обучения).

VI. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

VI.1. Основная литература (ОЛ)

1. Артемов А.В. Информационная безопасность [Электронный ресурс] / А.В. Артемов. _ Орел: МАБИВ, 2014. – 257 с. – Режим доступа: http://biblioclub.ru/index.php?

page=book&id=428605.
2. Загинайлов Ю.Н. Теория информационной безопасности и методология защиты информации [Электронный ресурс] / Ю.Н. Загинайлов. – М.; Берлин: Директ-Медиа, 2015. – 253 с. – Режим доступа: http://biblioclub.ru/index.php?page=book&id=276557.
3. Информационная безопасность и защита информации [Электронный ресурс]. – М.: Студенческая наука, 2012. - 1322 с. – Режим доступа: http://biblioclub.ru/index.php?page=book&id=227774.
4. Креопалов В.В. Технические средства и методы защиты информации [Электронный ресурс] / В.В. Креопалов. - М.: Евразийский открытый институт, 2011. – 278 с. – Режим доступа: URL http://biblioclub.ru/index.php?page=book&id=90753.
5. Петров А.А. Компьютерная безопасность. Криптографические методы защиты [Электронный ресурс]. – Саратов: Профобразование, 2017.– 446 с. – Режим доступа: http://www.iprbookshop.ru/63800.html.
6. Шаньгин В.Ф. Защита компьютерной информации. Эффективные методы и средства [Электронный ресурс]. – Саратов: Профобразование, 2017. – 544 с. – Режим доступа: http://www.iprbookshop.ru/63592.html.
7. Шаньгин В.Ф. Информационная безопасность и защита информации [Электронный ресурс] – Саратов: Профобразование, 2017. – 702 с. – Режим доступа: http://www.iprbookshop.ru/63594.html.

VI.2. Дополнительная литература (ДЛ)

8. Курячий Г.В. Операционная система Linux. Курс лекций [Электронный ресурс]. – Саратов: Профобразование, 2017. – 348 с. – Режим доступа: http://www.iprbookshop.ru/63944.html.
9. Электронное учебно-методическое пособие по дисциплине «Математические основы криптологии» / составители: Трунов И.Л., Поликарпов С.В. – Таганрог, 2015. – 30 с. http://open-edu.sfedu.ru/node/2755

VI.3. Перечень ресурсов сети Интернет

1. www.fstec.ru Официальный сайт ФСТЭК России.
2. www.tk26.ru Официальный сайт Технического комитета 26 России (по криптографии).
3. cve.mitre.org База знаний известных уязвимостей в области информационной безопасности.
4. www.nist.gov Официальный сайт Института стандартов и технологий США.
5. www.securitylab.ru Портал по информационной безопасности с ежедневными новостями из области информационной безопасности.
6. www.bugtraq.ru Портал по информационной безопасности с ежедневными новостями из области информационной безопасности.
7. www.opennet.ru Новостной портал по информационным технологиям.
8. www.habrhabr.ru Портал по информационным технологиям.
9. www.kaspersky.ru Официальный сайт российского разработчика антивирусных программ и сервисов.
10. www.drweb.com Официальный сайт российского разработчика антивирусных программ и сервисов.
11. www.intel.ru Официальный сайт компании Intel.
12. www.amd.ru Официальный сайт компании AMD.
13. www.cisco.com Официальный сайт компании Cisco Systems.
14. exelab.ru Портал, посвящённый исследованию программ
15. www.kali.org Сайт специального дистрибутива ОС Linux, предназначенного для

тестирования на проникновение.

16. www.metasploit.com Сайт средства разработки и исследования эксплойтов.
17. www.rutoken.ru Официальный сайт российского производителя аппаратных ключей и смарт-карт (RuToken).
18. www.aladdin-rd.ru Официальный сайт российского производителя аппаратных ключей и смарт-карт (Аладдин Р.Д.).
19. www.github.com Крупнейший ресурс открытого программного обеспечения.
20. www.eclipse.org Официальный сайт среды программирования Eclipse.
21. Галатенко В.А. Стандарты информационной безопасности: учебное пособие [Электронный ресурс]. – ИНТУИТ. – Режим доступа: http://www.intuit.ru/studies/courses/30/30/info.
22. Беляев А.В. Методы и средства защиты информации (курс лекций) [Электронный ресурс]. – Режим доступа: URL http://www.citforum.ru/internet/infsecure/index.shtml.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

VII.1. Учебно-лабораторное оборудование.

Аудитория К-322 оснащена: интерактивная доска SMART BOARD; персональный компьютер (9 шт.); ноутбук (4 шт); маршрутизатор (локальная сеть) (2 шт).

Аудитория И-419 содержит:

Персональные компьютеры (6 шт.); Стенд контроля ТЛФ линий; Стенд акустических и виброакустических измерений и активной защиты от утечки данных по виброакустическому каналу; Стенд поиска радиозакладок; Стенд для оценки ПЭМИН от радиоэлектронных устройств; Устройство «Пиранья» st031; Нелинейный локатор «Люкс»; Генератор высокочастотный Г4-116; Генератор высокочастотный Г4-109; Сканирующий приемник AR8020; Измерительная антенна ПДА5-0; Анализатор спектра GSP-810; Система виброакустического зашумления «Соната-АВ»; Генератор шума переносной SELSP-21B2; Лабораторный стенд видеосистемы наблюдения «Стилпост»; Лабораторный стенд видеосистемы наблюдения «Синергет»; Аппаратно-программный измерительный комплекс National Instruments.

VII.2. Программные средства:

1. Специально подготовленная версия операционной системы Linux (Ubuntu) с дополнительно установленным набором программ. Предназначена для выполнения лабораторных работ в лаборатории или на компьютере (ноутбуке) студента. Операционная система представлена в виде образа виртуальной машины для свободно распространяемой программы виртуализации Oracle VirtualBox, что позволяет легко создавать её копии, восстанавливать оригинальную версию, одновременно запускать на множестве компьютеров.

2. Программная модель под ОС Windows для проведения лабораторных работ по скремблированию речевых сигналов во временной и частотной областях. Модель позволяет производить сравнительный анализ эффективности различных способов скремблирования, а также исследования влияния различных параметров скремблирования (длина окна сигнала, количество блоков в окне, тип перестановки блоков в окне) на выходной сигнал.

VII.3. Технические и электронные средства:

В учебном процессе по дисциплине используется оборудование и программное обеспечение, указанные в п.7.1 и п.7.2. Для проведения лекционных занятий используются мультимедийные презентации лекций.

VIII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебный процесс обучения дисциплине включает в себя аудиторные занятия (лекции,

лабораторные работы) и самостоятельную работу. Промежуточная аттестация по дисциплине – дифференцированный зачёт. Лектор и преподаватели, ведущие занятия, контролируют посещение всех видов аудиторных занятий.

Лекционные занятия проводятся с обязательным использованием презентаций. Материалы лекционных занятий (презентации) предоставляются в пользование студентов посредством размещения в электронном пространстве (электронная система обучения lms.sfedu.ru).

Перед проведением лабораторных работ студенты должны быть ознакомлены под роспись с техникой безопасности при проведении работ в лаборатории. Перед выполнением лабораторных работ преподаватель даёт краткие пояснения по цели работы и процессу её выполнения. По результатам выполнения работы студенты оформляют отчёт о выполнении лабораторной работы с обязательным указанием названия работы, цели работы, используемого оборудования, хода работы и основных полученных результатов, в том числе обработки и анализа результатов лабораторной работы. Защита лабораторных работ осуществляется, как правило, на следующем занятии в форме собеседования.

Самостоятельная работа студентов включает в себя подготовку к лекционным, лабораторным занятиям и контрольным работам.

Текущий и рубежный контроль осуществляется на аудиторных занятиях при защите отчётов по лабораторным работам и на контрольных работах. Максимальное количество баллов по каждому виду контрольных мероприятий указано в учебной карте дисциплины.

Студенты, которые по уважительной причине не смогли набрать необходимое число баллов по текущему и рубежному контролю, могут по согласованию с преподавателем ликвидировать задолженности до конца последней недели теоретического обучения по дисциплине.

IX. УЧЕБНАЯ КАРТА ДИСЦИПЛИНЫ

Курс 2, семестр 3, очная форма обучения

№ п/п	Виды контрольных мероприятий (наименования оценочных средств)	Количество баллов	
		Текущий контроль	Рубежный контроль
	Модуль 1.		
1	Лабораторные работы №№ 1–4 (выполнение, подготовка отчёта, защита отчётов)	40 (4 работ × 10 баллов)	–
2	Контрольная работа №1	–	10
	Модуль 2.		
4	Лабораторные работы №№ 5–8 (выполнение, подготовка отчёта, защита отчётов)	40 (4 работ × 10 баллов)	–
5	Контрольная работа №2	–	10
Всего		**80**	**20**
Бонусные баллы		не предусмотрены	
Промежуточная аттестация в форме дифференцированного зачёта		Оценка по дисциплине выставляется по сумме баллов за текущий контроль и рубежный контроль: - 85–100 баллов – оценка «отлично»; - 71–84 балла – оценка «хорошо»; - 60–70 баллов – оценка «удовлетворительно»; - менее 60 баллов – оценка «неудовлетворительно»	

X. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

X.1. Паспорт фонда оценочных средств

№ п/п	Индикатор достижения компетенции	Наименование оценочного средства
1	**ОПК-2.1** Выбирает современные информационные технологии и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности.	- контрольные работы № 1-2.
2	**ОПК-2.3** Решает стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий	- лабораторные работы № 1-8; - контрольные работы № 1-2.
3	**ОПК-2.4** Учитывает основные требования информационной безопасности при решении стандартных задачи профессиональной деятельности	- контрольные работы № 1-2.
4	**ОПК-5.1** Инсталлирует и администрирует программное обеспечение на основе современных стандартов информационного взаимодействия систем	- лабораторная работа № 8;
5	**ОПК-5.3** Администрирует и осуществляет наладку аппаратного обеспечения информационных и автоматизированных систем	- лабораторные работы № 1-4;

X.2 Банк вопросов для составления контрольных работ

Раздел 1. **Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Критерии и стандарты защиты данных**

1. Что понимается под информационной безопасностью и защитой информации?
2. Чем отличаются понятия «информационная безопасность» и «компьютерная безопасность»?
3. Каковы основные составляющие информационной безопасности?
4. Что такое субъектно-объектная модель информационной системы?
5. Что понимается под уязвимостью информационной системы?
6. Что понимается под угрозой безопасности информации?
7. По каким критериям осуществляется классификация угроз безопасности информации?
8. Назовите основные методы реализации угроз информационной безопасности?
9. Что такое модель нарушителя?
10. Каковы основные причины, виды и каналы утечки информации?
11. Стандартизация в области информационной безопасности. Принципы и подходы.
12. Из каких частей состоит международный стандарт ISO/IEC 15408 и его отечественный аналог ГОСТ Р ИСО/МЭК 15408 и какие виды требований безопасности он содержит?
13. Что такое функциональные требования безопасности в ГОСТ Р ИСО/МЭК 15408.
14. Что такое требования доверия безопасности в ГОСТ Р ИСО/МЭК 15408.

Раздел 2. **Методы и средства аутентификации. Контроль доступа. Политики и модели безопасности**

1. Какова общая схема процедур идентификации/аутентификации?
3. Какие существуют способы аутентификации субъектов?
4. Какие требования предъявляются к парольным системам аутентификации субъекта, их преимущества и недостатки?
5. Каковы принципы формирования одноразовых паролей?
6. Как реализуется аутентификации субъектов с помощью одноразовых паролей?
7. Как реализуется аутентификация субъектов на основе модели «рукопожатия»?
8. Какие существуют типы персональных средств аутентификации субъектов, их основные особенности, преимущества и недостатки?
9. На какие два класса подразделяются биометрические средства аутентификации субъектов?
10. Какие биометрические признаки используются в статических биометрических системах аутентификации субъектов?
11. Принципы и особенности построения статических биометрических средств аутентификации субъектов, их преимущества, недостатки, сферы применения?
12. Какие биометрические признаки используются в динамических биометрических системах аутентификации субъектов?
13. Принципы и особенности построения динамических биометрических средств аутентификации субъектов, их преимущества, недостатки, сферы применения?
14. Как определяются ошибки биометрических средств аутентификации субъектов?
15. В чем суть понятий контроль доступа, авторизация?
16. В чем суть субъектно-объектной модели контроля доступа?
17. Что такое монитор безопасности, какие требования к нему предъявляются?

18. Что такое политика безопасности? В каком виде она может быть представлена?
19. Какие существуют основные виды политик безопасности?
20. Что такое модель безопасности. Какова общая схема модели безопасности?
21. Какие применяются модели безопасности?
22. Как реализуется дискреционная модель безопасности?
23. Как реализуется мандатная модель безопасности?
24. Как реализуется ролевая модель безопасности?

Раздел 3. **Техническая защита информации**

1. Какие технические каналы утечки информации возникают при обработке информации средствами вычислительной техники?
2. Электромагнитные каналы утечки информации?
3. Электрические каналы утечки информации?
4. Параметрический канал утечки информации?
5. Технические каналы утечки акустической (речевой) информации?
6. Воздушные технические каналы утечки информации?
7. Вибрационные технические каналы утечки информации?
8. Электроакустические технические каналы утечки информации?
9. Оптико-электронный технический канал утечки информации?
10. Какие существуют портативные средства технической разведки?
11. Портативные средства акустической разведки?
12. Портативные средства радио-, радиотехнической разведки?
13. Средства компьютерного шпионажа?
14. Портативные средства съема информации с проводных линий связи?
15. Оперативные средства видеонаблюдения и съемки?
16. Какие применяются методы и средства защиты от технической разведки?
17. Какие существуют способы и аппаратура защиты телефонных линий?
18. Какие существуют средства защиты от НСД к акустической информации?
19. Какие существуют средства обнаружения радиозакладок?

Раздел 4. **Организационно-правовое обеспечение информационной безопасности**

1. Место информационного права в общей системе права?
2. Какие существуют правовые и организационные методы защиты информации?
3. Как выглядит классификация видов информационных ресурсов?
4. Как классифицируется информационные ресурсы по категориям доступа?
5. Понятие государственной тайны и принципы ее защиты?
6. Каковы основные принципы защиты коммерческой тайны?
7. Каковы основные принципы защиты профессиональной и служебной тайны?
8. Какие существуют виды профессиональной тайны?
9. Что составляет правовую основу системы лицензирования и сертификации?
10. Как осуществляется лицензирование деятельности по защите государственной тайны?
11. Каковы основные принципы сертификации средств защиты информации?
12. Как осуществляется лицензирование и сертификация в области защиты конфиденциальной информации?
13. Каковы особенности правового обеспечения защиты информации в ИС?

15. Понятие киберпреступности?
16. Основные проблемы борьбы с киберпреступностью?
17. Классификация способов совершения компьютерных преступлений?
18. Виды программно-технических экспертиз для доказательства киберпреступлений?

Раздел 5. **Криптографические методы защиты информации и криптоанализ**

1. Атаки на криптографические системы и алгоритмы: поясните понятия криптоатака и криптостойкость; информация, доступная злоумышленнику при анализе шифра; основные показатели криптостойкости; приведите и поясните основные методы криптоанализа шифров.
2. Поточные шифры: принципы шифрования и расшифрования информации при использовании гаммы (ключевой псевдослучайной последовательности), основные требования к псевдослучайным последовательностям).
3. Блочные шифры: принципы шифрования и расшифрования информации; режимы функционирования блочных шифров; возможности, которыми обладает злоумышленник при анализе блочного шифра; основные требования к блочному шифру.
4. Российский стандарт блочного шифрования «Кузнечик»: основные параметры, структура раунда (цикла) шифрования и принцип действия, процедура разворачивания исходного ключа в раундовые (рабочие) подключи, режимы использования.
5. Асимметричные криптографические системы: причины появления и основные идеи; односторонние функции; факторизация; дискретный логарифм и дискретный корень.
6. Криптографический алгоритм RSA: математическая проблема, лежащая в основе алгоритма; принцип формирования открытого и закрытого ключей пользователя; принцип шифрования сообщений; привести пример шифрования сообщения.
7. Криптографическая система Эль-Гамаля: математическая проблема, лежащая в основе алгоритма; принцип формирования открытого и закрытого ключей пользователя; принцип шифрования сообщений.
8. Хеш-функции: назначение хеш-функций; основные требования к хеш-функциям; привести краткое описание функции хеширования SHA-3.
9. Хеш-функции: назначение хеш-функций; основные требования к хеш-функциям; привести краткое описание функции хеширования GOST R 34.11-2012 (Streebog).
10. Электронная подпись: назначение; основные нарушения аутентичности сообщений; электронная подпись на основе алгоритма RSA.
11. Управление ключами: назначение; поясните такие компоненты, как генерация ключей, накопление ключей и распределение ключей; привести краткое описание схемы открытого распределения ключей Диффи-Хеллмана.
12. Протоколы аутентификации с симметричными алгоритмами шифрования (В чем заключается идея строгой аутентификации? Приведите и поясните следующие методы: односторонняя аутентификация, основанная на метках времени; односторонняя аутентификация, основанная на использовании случайных чисел; двусторонняя аутентификация, использующая случайные значения).

Раздел 6. **Вредоносное программное обеспечение**

1. Что такое вредоносное ПО, разновидности, цели и задачи его создания и применения?
2. Какие существуют методы защиты от вредоносного ПО?

3. Какие дефекты ОС могут привести к созданию каналов утечки информации? Приведите и поясните основные уязвимости программного обеспечения.
4. Поясните метод выявления вредоносных программ на основе сигнатурного поиска, приведите его достоинства и недостатки.
5. Поясните метод выявления вредоносных программ на основе эвристического поиска, приведите его достоинства и недостатки.
6. Поясните метод выявления вредоносных программ на основе мониторинга информационных потоков, приведите его достоинства и недостатки.
7. Поясните метод выявления вредоносных программ на основе контроля целостности, приведите его достоинства и недостатки.
8. Поясните метод выявления вредоносных программ на основе программных ловушек, приведите его достоинства и недостатки.

Раздел 7. **Безопасность операционных систем**

1. Понятие защищённой операционной системы (Типичные атаки на операционные системы. Что такое защищённая операционная система, политика безопасности и адекватная политика безопасности? Приведите основные подходы к построению защищённых операционных систем.)
2. Задачи аппаратного обеспечения защиты информации в операционных системах. (Приведите основные подходы к обеспечению защиты оперативной памяти. Для чего предназначено планирование задач в многозадачной ОС? Каким образом осуществляется защита сегментов оперативной памяти в процессорах семейства x86 и x86-64?)
3. Подсистема аудита операционной системы. (Общие сведения об аудите. Чем вызвана необходимость включения в операционную систему функций аудита? Требования к подсистеме аудита.)
4. Шифрование файлов в Windows при использовании Encrypted File System (EFS). (Приведите основные причины ненадёжности криптографических систем, дайте краткое пояснение этих причин. Каким образом осуществляется шифрование файлов в ОС Windows при использовании EFS?)
5. Шифрование в популярных архиваторах. (Приведите основные причины ненадёжности криптографических систем при использовании в архиваторах, дайте краткое пояснение этих причин. Каким образом осуществляется шифрование в популярных архиваторах? Какой стойкостью обладают указанные версии архиваторов?)
6. Программные средства прозрачного шифрования дисков (Какие возможности предоставляют программные средства прозрачного шифрования дисков? Приведите причины возможной ненадёжности программных средств прозрачного шифрования дисков. Каким образом можно нейтрализовать данные причины?).
7. Привести краткое описание и характеристики расширения команд AES-NI и NX-Bit (ND-Bit).
8. Системы разграничения доступа на основе технологии виртуализации. (Приведите основные виды технологий виртуализации, их достоинства и недостатки.)
9. Технология рандомизации расположения адресного пространства ASLR. (Назначение, способы обхода защиты, достоинства и недостатки).

Раздел 8. **Защита информации в компьютерных сетях**

1. Основные функции защиты информации в сетях передачи данных (поясните следующие

компоненты: конфиденциальность, аутентификация, целостность, невозможность обмана, управление доступом, доступность ресурсов).

2. Функции межсетевых экранов (Где должен устанавливаться МЭ? Основные задачи, решаемые МЭ. По каким признакам можно классифицировать МЭ? Поясните следующие функции МЭ – фильтрация трафика и выполнение функций посредничества).
3. Экранирующий маршрутизатор (назначение, принцип работы, достоинства и недостатки).
4. Шлюз сеансового уровня (назначение, принцип работы, достоинства и недостатки).
5. Шлюз прикладного уровня (назначение, принцип работы, достоинства и недостатки).
6. Критерии оценки качества межсетевых экранов (Каким группам требований должны удовлетворять МЭ? Приведите основные классы защищенности МЭ в соответствии с руководящими документами ФСТЭК России. Каким требованиям должны удовлетворять МЭ в соответствии с классами защищенности?)
7. Виртуальные защищенные сети VPN. (Что такое виртуальная защищенная сеть? Принцип механизма туннелирования, критерии безопасности передаваемых данных и способы их обеспечения.)
8. Протокол защиты сетевого уровня IPSec (назначение, особенности средств VPN данного уровня, используемые криптографические технологии, основные компоненты, достоинства и недостатки).
9. Протоколы защиты сеансового уровня SSL/TLS (назначение, особенности средств VPN данного уровня, используемые криптографические технологии, основные этапы формирования и поддержания защищаемого соединения, достоинства и недостатки).
10. Протоколы защиты канального уровня L2F и L2TP (назначение, особенности средств VPN данного уровня, используемые криптографические технологии, основные этапы формирования и поддержания защищаемого соединения, достоинства и недостатки).
11. Адаптивное управление безопасностью сети (предпосылки использования; пояснить следующие основные компоненты – оценка риска, анализ защищенности, обнаружение атак; поясните назначение и принцип действия сканеров безопасности).
12. Системы обнаружения атак (назначение систем обнаружения атак; приведите и поясните методы анализа сетевой информации – статистический метод, экспертные системы, нейронные сети; достоинства и недостатки указанных методов).

X.3. Контрольная работа по модулю 1

Контрольная работа по модулю 1 предназначена для проведения рубежного контроля по разделам 1-4: «Основные понятия и определения. Уязвимости информационных систем. Угрозы безопасности информации. Критерии и стандарты защиты данных»; «Методы и средства аутентификации. Контроль доступа. Политики и модели безопасности»; «Техническая защита информации» и «Организационно-правовое обеспечение информационной безопасности». Контрольная работа позволяет оценить уровень сформированных знаний, умений и навыков в рамках формируемых указанными разделами дисциплины компетенции, в том числе в ходе самостоятельной работы обучающегося.

Контрольная работа проводится в письменной форме. При написании ответа обучающемуся необходимо ответить на 4 контрольных вопроса. Каждый из 4 вопросов контрольной работы выбирается (случайным образом) из банка вопросов соответствующего раздела: первый вопрос – из банка вопросов по разделу 1, второй вопрос – из банка вопросов по разделу 2 и т. д.

Пример содержания билета для контрольной работы по модулю 1:

Контрольная работа №1
по дисциплине «Безопасность информационных технологий»

1. Что понимается под информационной безопасностью и защитой информации?
2. Какова общая схема процедур идентификации/аутентификации?
3. Какие технические каналы утечки информации возникают при обработке информации средствами вычислительной техники?
4. Место информационного права в общей системе права?

Критерии оценивания:

Контрольная работа оценивается в максимальное количество баллов, равное 10, что соответствует правильным ответам на все вопросы. Правильный ответ на каждый из 4 вопросов оценивается максимум в 2,5 балла, значение балла может уменьшаться пропорционально полноте ответу или количеству правильных ответов на подвопросы. После суммирования баллов по всем правильным ответам набранная сумма баллов округляется до целого значения.

X.4. Контрольная работа по модулю 2

Контрольная работа по модулю 1 предназначена для проведения рубежного контроля по разделам 5-8: «Криптографические методы защиты информации и криптоанализ»; «Безопасность операционных систем»; «Вредоносное программное обеспечение» и «Защита информации в компьютерных сетях». Контрольная работа позволяет оценить уровень сформированных знаний, умений и навыков в рамках формируемых указанными разделами дисциплины компетенции, в том числе в ходе самостоятельной работы обучающегося.

Контрольная работа проводится в письменной форме. При написании ответа обучающемуся необходимо ответить на 4 контрольных вопроса. Каждый из 4 вопросов контрольной работы выбирается (случайным образом) из банка вопросов соответствующего раздела: первый вопрос – из банка вопросов по разделу 5, второй вопрос – из банка вопросов по разделу 6 и т. д.

Пример содержания билета для контрольной работы по модулю 2:

Контрольная работа №2
по дисциплине «Безопасность информационных технологий»

1. Поточные шифры: принципы шифрования и расшифрования информации при использовании гаммы (ключевой псевдослучайной последовательности), основные требования к псевдослучайным последовательностям).
2. Что такое вредоносное ПО, разновидности, цели и задачи его создания и применения?
3. Понятие защищённой операционной системы (Типичные атаки на операционные системы. Что такое защищённая операционная система, политика безопасности и адекватная политика безопасности? Приведите основные подходы к построению защищённых операционных систем.)

4. Функции межсетевых экранов (Где должен устанавливаться МЭ? Основные задачи, решаемые МЭ. По каким признакам можно классифицировать МЭ? Поясните следующие функции МЭ – фильтрация трафика и выполнение функций посредничества).

Критерии оценивания:

Контрольная работа оценивается в максимальное количество баллов, равное 10, что соответствует правильным ответам на все вопросы. Правильный ответ на каждый из 4 вопросов оценивается максимум в 2,5 балла, значение балла может уменьшаться пропорционально полноте ответу или количеству правильных ответов на подвопросы. После суммирования баллов по всем правильным ответам набранная сумма баллов округляется до целого значения.

X.5. Лабораторные работы по модулю 1

Лабораторная работа 1.

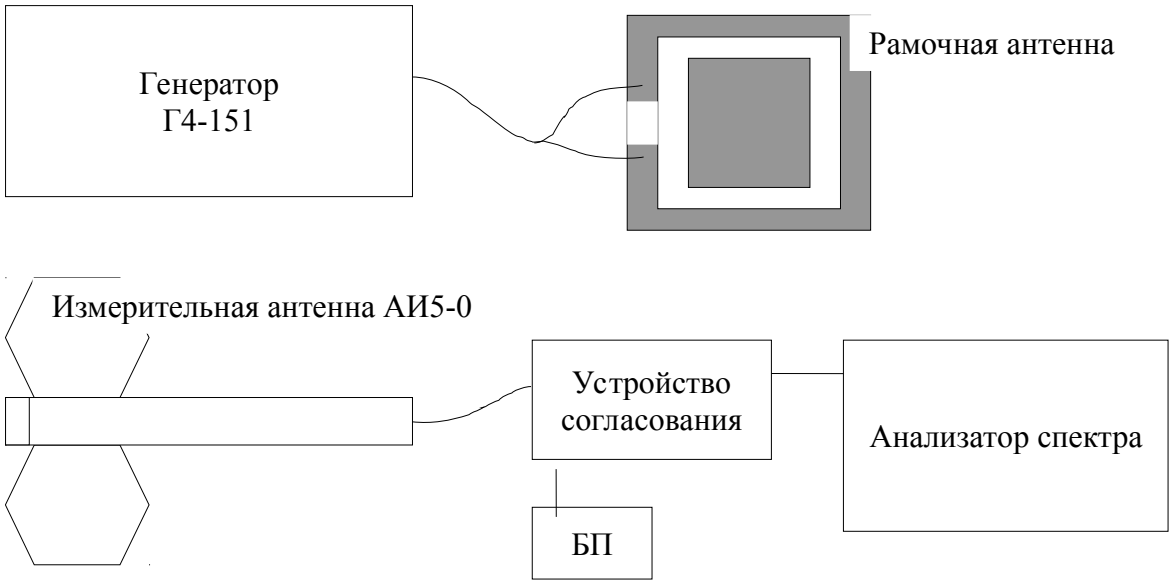
«ИССЛЕДОВАНИЕ КАНАЛОВ УТЕЧКИ ИНФОРМАЦИИ ЗА СЧЕТ ПОБОЧНЫХ ЭЛЕКТРОМАГНИТНЫХ ИЗЛУЧЕНИЙ»

Цель работы. Определение количественных характеристик электромагнитных излучений радиоэлектронного оборудования (измерительных приборов, компьютеров, бытовых радиоприемников, пейджеров и т. п.).

Объект исследования. Генератор стандартных сигналов Г4-151.

Инструментарий для исследования. Измерения проводятся при помощи анализатора спектра системы LabVIEW. Генератор Г4-151 дополняется излучающей антенной рамочного типа. Спектроанализатор снабжается приемной измерительной антенной АИ5-0.

Содержание исследований. Проводятся измерения уровней излучения генератора на нескольких частотах. Вначале на генераторе устанавливается максимальная мощность излучения, отключается излучающая антенна и делается попытка обнаружить сигнал. Затем к генератору подключается рамочная антенна и проводятся измерения на нескольких частотах. Для частоты, на которой обнаружен максимальный уровень излучения, необходимо провести измерение уровней излучения на разных расстояниях и определить максимальную дальность обнаружения излучений прибора. По результатам измерений необходимо построить зависимость уровня сигнала излучения от расстояния до излучателя.



Лабораторная работа 2.

«МЕТОДЫ СЪЁМА ИНФОРМАЦИИ ПО ВИБРОАКУСТИЧЕСКОМУ КАНАЛУ И МЕРЫ ПРОТИВОДЕЙСТВИЯ УТЕЧКЕ ИНФОРМАЦИИ ПО ВИБРОАКУСТИЧЕСКОМУ КАНАЛУ»

Цель работы. Изучить распространение звуковых колебаний по металлической конструкции, определить скорость распространения звука импульсным методом.

Объект исследования. Виброакустический канал утечки информации

Инструментарий для исследования. Конструкция для передачи виброакустических колебаний в виде отрезка металлического профиля. Устройство ввода звука в конструкцию – виброакустический преобразователь ВИ-45, подключённый к виртуальному генератору

сигналов низкой частоты gen3. Устройство ввода заградительного вибрационного шума – виброакустический преобразователь ВИ-45, подключенный к генераторному блоку “СОНАТА-АВ”. Выносной датчик виброакустического приёмника многофункционального поискового прибора ST-031P “ПИРАНЬЯ” и зажим для его крепления.

Содержание исследований. Необходимо получить совместную АЧХ тракта генерации и приёма виброакустического сигнала (датчик на излучателе), тракта передачи сигнала на датчик на конце строительной конструкции. Далее выполнить те же измерения при включённой системе виброакустического зашумления СОНАТА-АВ.

Порядок проведения работы:

1. При выключенном генераторном блоке системы акустической защиты ”СОНАТА” построить АЧХ тракта передачи сигнала на основе 20 точек в полосе частот от 300 до 6000 Гц. Для этого:

a. Включить генератор низкой частоты gen3 с усилителем мощности и с подключенным к нему виброакустическим излучателем, установить частоту генерации в 300 Гц и уровень выходного сигнала 10 В. При нормальной работе вблизи излучателя должен прослушиваться звуковой тон.

b. Включить многофункциональный поисковых прибор ST-031P с подключенным выносным виброакустическим датчиком, установить режим осциллографа, обеспечить механический контакт рабочей поверхности датчика с боковой поверхностью конструкции в непосредственной близости от излучателя 1 при помощи зажима. При этом на ЖК-экране прибора ST-031P должна появиться осциллограмма сигнала синусоидальной формы.

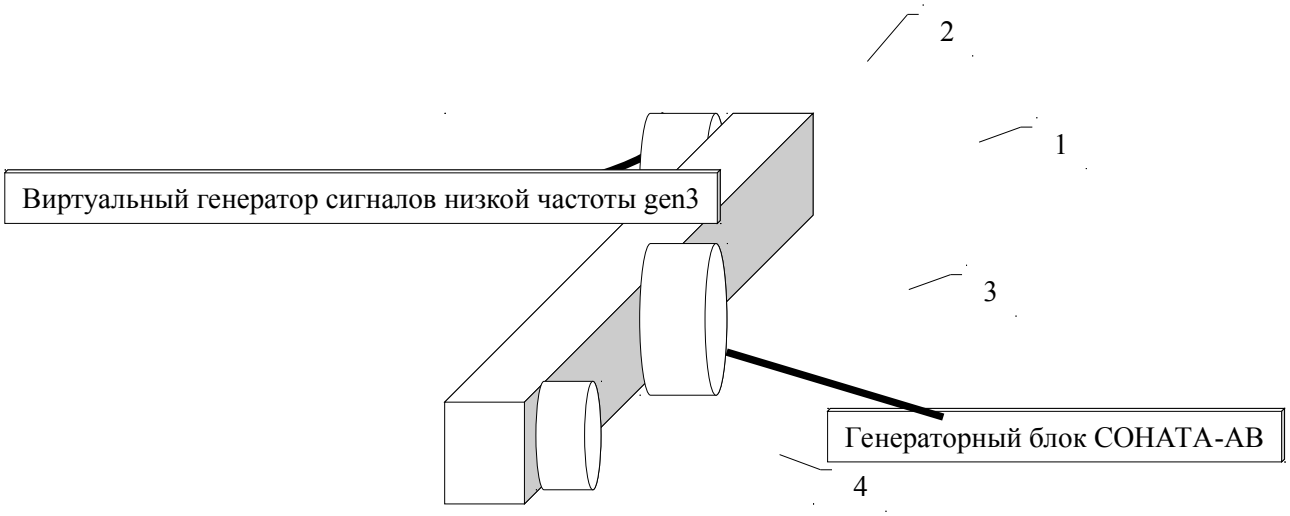
c. После установления синхронизации и фиксации фазы осциллограммы при помощи регулятора выходного уровня генератора Г3-56 установить максимальный уровень сигнала, при котором не возникает клиппирования (визуального ограничения считываемой осциллограммы по амплитуде).

d. Изменяя частоту на выходе генератора от 300 до 6000 Гц убедиться, что ни при одной частоте не возникает клиппирование и перегрузка входного канала осциллографа (надпись “OVR” по центру экрана).

e. Установить частоту генерируемого сигнала 300 Гц, и записать показания амплитуды сигнала, индицируемые на ЖК-дисплее ST-031P. С шагом из диапазона 200-500 Гц (задаётся преподавателем) изменять частоту до верхнего предела 6000 Гц. Записать значения частот и амплитуд в таблицу 1. По результатам измерений построить график.

f. Повторить измерения с датчиком, установленным на конструкции на максимальном расстоянии от излучателя, построить зависимость ослабления звука.

2. Включить генераторный блок “СОНАТА-АВ”, установить значение уровня генерируемого сигнала на минимальное значение, режим “Б” работы первого канала зашумления (нагрузка на виброакустические излучатели). Повторить выполнение пункта (e) при излучателе, установленном между излучателями 2 и 3.



Лабораторная работа 3.
«СКРЕМБЛЕР РЕЧЕВОГО СИГНАЛА»

Цель работы. Изучение работы программной модели скремблера. Данная программная модель позволяет проводить скремблирование речевых сигналов с помощью различных способов: скремблирование во временной и частотной областях, а также производить сравнительный анализ того, какой из способов скремблирования дает наилучший результат (наименее узнаваемый сигнал). Также предоставляется возможность исследования влияния различных параметров скремблирования (длина окна сигнала, количество блоков в окне, тип перестановки блоков в окне) на выходной сигнал

Объект исследования. Скремблер речевого сигнала.

Инструментарий для исследования. В качестве исходных сигналов скремблера используются звуковые файлы в формате WAV, записанные с частотой дискретизации 8 КГц в формате моно. Реализация программной модели выполнена на языке ObjectPascal в системе визуального программирования Borland Delphi 4.0 фирмы Inprise. Скремблер может выполнять загрузку звуковых файлов в формате WAV, их скремблирование или дескремблирование по одному из нескольких алгоритмов с изменяемыми параметрами, запись результата в файлы в формате WAV, а также визуализировать и озвучить как исходный, так и обработанный звук.

Порядок проведения работы. Загрузить звуковой файл и провести скремблирование (дескремблирование) загруженного файла при различных длинах блоков на которые разбивается данный звуковой файл Провести экспертную оценку на слух степени искаженности полученного при скремблировании речевого сигнала для различных скремблеров и при различных размеров блоков разбиения.

Лабораторная работа 4.
«ПОИСК НЕОДНОРОДНОСТЕЙ В КАБЕЛЬНЫХ ЛИНИЯХ»

Цель работы. Изучение способа обнаружения и локализации неоднородностей в кабельных линиях на основе рефлектометрического метода.

Инструментарий для выполнения лабораторной работы. Проведение рефлектометрических измерений осуществляется на основе технологии виртуальных измерительных приборов реализованной в аппаратно-программном комплексе National Instruments. Для реализации метода рефлектометрии используются

виртуальные приборы “Генератор” и “Осциллограф” на основе аппаратного генератора сигнала произвольной формы NI PXI-5412 и высокочастотного цифрового преобразователя NI PXI-5152.

Выполнение работы. Необходимо соединить выход "Генератора" со входом осциллографа через тройник, ко второму входу тройника подсоединить отрезок эталонного кабеля известной длины. Запустить виртуальные генератор и осциллограф. Определить время распространения импульса по кабелю. На экране дисплея отобразятся исходный пик от генератора и отраженный пик от конца эталонного кабеля. Для этого подвести курсоры к вершинам пиков на экране дисплея и по таблице, появившейся на экране дисплея определить время распространения. Вычислить скорость прохождения сигнала по кабелю эталонного отрезка исходя из известной длины отрезка. Подсоединить неизвестный кабель вместо эталонного и определить с помощью курсоров время распространения сигнала по кабелю с помощью процедуры. аналогичной описанной выше. Определить длину кабеля. К оконечному разъему исследуемого кабеля подключить устройство имитирующую емкостную и резистивную нагрузки различных номиналов. Сделать скриншоты полученных осциллограмм и внести данные в отчет.

Требования к содержанию и оформлению отчётов о выполнении лабораторных работ:

- отчёт должен содержать титульный лист с указанием названия работы, фамилий выполнивших студентов, номера группы (подгруппы, бригады), фамилии преподавателя; цель лабораторной работы; перечень используемого в лабораторной работе оборудования; ход работы со всеми полученными результатами измерений в текстовом или графическом виде; обработку результатов измерений в соответствии с требованиями задания на выполнение лабораторной работы, выводы по лабораторной работе;
- отчёт должен быть оформлен в соответствии с требованиями к оформлению текстовых документов (ГОСТ 7.32-2001, ГОСТ 2.105-95).

Критерии оценивания лабораторных работ

Согласно учебной карте дисциплины на лабораторные работы по модулю 1 отводится 40 баллов. Каждая лабораторная работа оценивается в максимум 10 баллов. При оценивании лабораторной работы принимается во внимание активное участие в выполнении лабораторной работы; соблюдение требований к содержанию и оформлению отчёта о выполнении лабораторной работы, ответы на вопросы в процессе защиты отчёта о выполнении лабораторной работы.

Условием допуска к защите отчёта о выполнении лабораторной работы является факт выполнения задания лабораторной работы (в рамках аудиторных лабораторных занятий). Обучающийся, не выполнявший лабораторную работу, получает по ней 0 (ноль) баллов.

Оценивание каждой лабораторной работы проводится во время процедуры защиты отчёта о выполнении работы. Оценивание каждой лабораторной работы (в пределах 10 баллов) осуществляется по следующим элементам оценивания:

- до 4 баллов – оценивание выполнения работы и пояснений обучающимся хода выполнения работы;
- до 4 баллов – оценивание отчёта о выполнении лабораторной работы и пояснений обучающимся содержимого отчёта о выполнении лабораторной работы;
- до 2 баллов – оценивание ответа обучающегося на вопросы, связанные с

принципами работы используемых в лабораторной работе средств.

Допускается каждый элемент оценивания лабораторной работы оценивать дробным числом баллов; после суммирования баллов по всем элементам оценивания набранная сумма баллов округляется до целого значения.

Критерии оценивания выполнения работы и пояснений обучающимся хода выполнения работы:

4 балла – обучающийся принимал активное участие в выполнении работы, может дать пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы);

2,1-3,9 балла – обучающийся принимал участие в выполнении работы, может дать частичные пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы); ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-2,0 балл – обучающийся принимал участие в выполнении работы, может дать частичные пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы); обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – обучающийся не может дать даже частичные пояснения хода выполнения работы, несмотря на участие в выполнении работы (присутствии на лабораторном занятии).

Критерии оценивания отчёта о выполнении лабораторной работы и пояснений обучающимся содержимого отчёта о выполнении лабораторной работы:

4 балла – отчёт о выполнении работы содержит все требуемые разделы; отчёт соответствует заданному варианту; отчёт оформлен в соответствии с требованиями к оформлению текстовых документов (ГОСТ 7.32-2001, ГОСТ 2.105-95); отчёт содержит все требуемые результаты измерений; отчёт содержит все требуемые обработки результатов измерений; обучающийся может дать полные пояснения любого места отчёта о выполнении лабораторной работы, в том числе полученных результатов;

2,1-3,9 балла – отчёт о выполнении работы содержит все требуемые разделы; отчёт соответствует варианту выполнения экспериментальных исследований; отчёт оформлен в соответствии с требованиями к оформлению текстовых документов; отчёт содержит все требуемые результаты измерений; отчёт содержит большинство требуемых обработок результатов измерений; обучающийся может дать частичные пояснения отдельных фрагментов отчёта о выполнении лабораторной работы, в том числе использованных процедур обработки измерений и полученных результатов; ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-2,0 балл – отчёт о выполнении работы содержит основные требуемые разделы (в том числе ход работы и обработку результатов); отчёт соответствует варианту выполнения экспериментальных исследований; отчёт в целом оформлен в соответствии с требованиями к оформлению текстовых документов без грубых нарушений требований; отчёт содержит большинство требуемых результатов измерений; отчёт содержит большинство требуемых обработок результатов измерений; обучающийся может дать частичные пояснения отдельных фрагментов отчёта о выполнении лабораторной работы, в том числе использованных процедур обработки измерений и полученных результатов; обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – либо отчёт о выполнении работы не содержит все требуемые разделы; либо отчёт не соответствует варианту выполнения экспериментальных исследований; либо отчёт оформлен с грубыми нарушениями требований к оформлению текстовых документов; либо отчёт не содержит большинства требуемых результатов измерений; либо отчёт не содержит большинства требуемых обработок результатов измерений; либо обучающийся не может дать даже частичные пояснения отдельных фрагментов отчёта о выполнении лабораторной работы, в том числе использованных процедур обработки измерений и полученных результатов.

Критерии оценивания ответа обучающегося на вопросы, связанные с принципами проведения соответствующих видов измерений и принципами работы используемых в лабораторной работе средств измерений:

2 балл – обучающийся может дать полные ответы на вопросы, связанные с принципами проведения изучаемых в лабораторной работе видов измерений, и на вопросы, связанные с принципами работы используемых в лабораторной работе средств измерений;

1-1,9 балла – обучающийся может дать частично верные ответы на вопросы, связанные с принципами проведения изучаемых в лабораторной работе видов измерений, и (или) на вопросы, связанные с принципами работы используемых в лабораторной работе средств измерений; ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-0,9 балла – обучающийся может дать частично верные ответы на вопросы, связанные с принципами проведения изучаемых в лабораторной работе видов измерений, и (или) на вопросы, связанные с принципами работы используемых в лабораторной работе средств измерений; обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – обучающийся не может дать даже частично верные ответы ни на вопросы, связанные с принципами проведения изучаемых в лабораторной работе видов измерений, ни на вопросы, связанные с принципами работы используемых в лабораторной работе средств измерений.

X.6. Лабораторные работы по модулю 2

Лабораторная работа №5
«ВОССТАНОВЛЕНИЕ И ФОРМИРОВАНИЕ ПАРОЛЕЙ ПОЛЬЗОВАТЕЛЕЙ»

Цель лабораторной работы:

1) закрепить навыки обучающихся по правильному формированию паролей пользователей путём сравнения эффективности восстановления паролей, имеющих различные параметры;

2) показать, как неправильное использование стойких криптографических алгоритмов может привести к созданию слабозащищённых систем идентификации и аутентификации пользователей.

В ходе выполнения данного лабораторной работы обучающийся получит навыки по самостоятельному формированию различных хэш-функций от паролей пользователей, получит возможность сравнить эффективность двух различных подходов к восстановлению паролей пользователей: на основе методов перебора (полного, по словарю, с мутациями символов и т.д.) и на основе техники криптоанализа по размену «время — память» (с использованием радужных таблиц).

1. Описание RainbowCrack

RainbowCrack является инструментом для восстановления по хэшам исходных открытых сообщений, в роли которых нередко выступают пароли пользователей. Тогда как традиционные методы силового восстановления подразумевают последовательный перебор исходных открытых сообщений с последующим сравнением полученных хэшей, RainbowCrack работает другим способом. Он осуществляет особым образом предварительное вычисление всех возможных пар «открытый текст — зашифрованный текст» и сохраняет их в файле, называемом "rainbow table". Предвычисление таких таблиц занимает длительное время, но по завершении этих предвычислений появляется возможность восстановить открытое сообщение (пароль) в течении секунд. Таким образом осуществляется «размен время — память».

RainbowCrack поддерживает работу с несколькими хэш-функциями: md5, sha1 и др. Неподдерживаемые хэш-функции можно достаточно легко добавить самому, так как данная программа имеет открытые исходные коды.

RainbowCrack работает как под управлением ОС Windows, так и ОС Линукс. Самое главное, в состав программы включена утилита для генерации rainbow-таблиц, что позволяет формировать таблицы для различных наборов символов, например, для русского языка и т.д.

Разработчиками программы предлагается несколько готовых конфигурация для rainbow-таблиц.

Конфигурация #0	
Хэш алгоритм	lm
Набор символов	alpha (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
Диапазон длин открытых сообщений	1 - 7
Объём пространства ключа	26^1 + 26^2 + 26^3 + 26^4 + 26^5 + 26^6 + 26^7 = 8353082582
t	2100
m	8000000
l	5

Использование диска	m * 16 * l = 640000000 B = 610 MB
Вероятность успеха	0.9990
Среднее время криптоанализа	3.7841 s
Среднее время криптоанализа на системах с малым объёмом памяти (свободной памяти менее, чем 122MB)	8.2836 s
Максимальное время криптоанализа	31.1441 s
Команды, используемые для предвычисления таблиц	rtgen lm alpha 1 7 0 2100 8000000 all rtgen lm alpha 1 7 1 2100 8000000 all rtgen lm alpha 1 7 2 2100 8000000 all rtgen lm alpha 1 7 3 2100 8000000 all rtgen lm alpha 1 7 4 2100 8000000 all
Длительность процедуры предвычисления таблиц	2 days 18 hours

Конфигурация #1	
Хэш алгоритм	lm
Набор символов	alpha-numeric(ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789)
Диапазон длин открытых сообщений	1 - 7
Объём пространства ключа	36^1 + 36^2 + 36^3 + 36^4 + 36^5 + 36^6 + 36^7 = 80603140212
t	2400
m	40000000
l	5
Использование диска	m * 16 * l = 3200000000 B = 3 GB
Вероятность успеха	0.9904
Среднее время криптоанализа	7.6276 s
Среднее время криптоанализа на системах с малым объёмом памяти (свободной памяти менее, чем 122MB)	13.3075 s
Максимальное время криптоанализа	40.6780 s
Команды, используемые для предвычисления таблиц	rtgen lm alpha-numeric 1 7 0 2400 40000000 all rtgen lm alpha-numeric 1 7 1 2400 40000000 all rtgen lm alpha-numeric 1 7 2 2400 40000000 all rtgen lm alpha-numeric 1 7 3 2400 40000000 all rtgen lm alpha-numeric 1 7 4 2400 40000000 all
Длительность процедуры предвычисления таблиц	15 days 17 hours

2. Порядок работы с Rainbow_crack

Допустим, мы скопировали (установили) Rainbowcrack в каталог «cd c:\rainbowcrack-1.2-win». После этого запускаем консоль Windows (команда cmd) и осуществляем переход в данный каталог используя команды «**cd ..**» и «**cd c:\rainbowcrack-1.2-win**».

Далее, мы можем воспользоваться командами данного программного продукта.

Для начала необходимо подготовить радужные таблицы (rainbow table). Для этого служит команда «**rtgen.exe**». Вводим её в командной строке и нажимаем «**Enter**». В этом случае на

экран будет выведено следующее сообщение, подсказывающее, как пользоваться командой:

```
C:\rainbowcrack-1.2-win>rtgen
RainbowCrack 1.2 - Making a Faster Cryptanalytic Time-Memory Trade-Off
by Zhu Shuanglei <shuanglei@hotmail.com>
http://www.antsight.com/zsl/rainbowcrack/

usage: rtgen hash_algorithm \
plain_charset plain_len_min plain_len_max \
rainbow_table_index \
rainbow_chain_length rainbow_chain_count \
file_title_suffix
rtgen hash_algorithm \
plain_charset plain_len_min plain_len_max \
rainbow_table_index \
-bench

hash_algorithm: available: lm md5 sha1
plain_charset: use any charset name in charset.txt here
use "byte" to specify all 256 characters as the charset of the plaintext
plain_len_min: min length of the plaintext
plain_len_max: max length of the plaintext
rainbow_table_index: index of the rainbow table
rainbow_chain_length: length of the rainbow chain
rainbow_chain_count: count of the rainbow chain to generate
file_title_suffix: the string appended to the file title
add your comment of the generated rainbow table here
-bench: do some benchmark

example: rtgen lm alpha 1 7 0 100 16 test
rtgen md5 byte 4 4 0 100 16 test
rtgen sha1 numeric 1 10 0 100 16 test
rtgen lm alpha 1 7 0 -bench

C:\rainbowcrack-1.2-win>
```

Допустим, нам требуется создать радужную таблицу для хэш-функции SHA1, при этом удовлетворяющую условиям:

1) восстанавливаемые пароли содержат только маленькие латинские буквы и цифры;

2) длина восстанавливаемых паролей от 1-го до 6-и символов;

3) название таблицы — laba.

Для определения того, как указать используемый набор символов, просмотрим содержимое файла «charset.txt»:

```
# charset configuration file for rainbowcrack 1.1 and later
# by Zhu Shuanglei <shuanglei@hotmail.com>

alpha = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-numeric = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
alpha-numeric-symbol14 = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
all = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]

numeric = [0123456789]
loweralpha = [abcdefghijklmnopqrstuvwxyz]
loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]
```

```
lowerrus-numeric = [ёйцукенгшщзхъфывапролджэячсмитьбю0123456789]
```

Как видно из содержимого данного файла, необходимый нам набор символов называется «loweralpha-numeric». В принципе, мы можем сами добавлять в этот файл различные наборы символов, например «lowerrus-numeric».

Следующим важным шагом является определение размерности таблицы, которая зависит от объёма пространства входных сообщений. Данный объём легко вычислить, зная основание алфавита и количетсво используемых символов. Например, набор «loweralpha-numeric» содержит 36 различных символов (т.е. основание равно 36). Для паролей длиной 1 символ объём пространства составляет 36^1 или 36 различных комбинаций, для паролей длиной 2 символа объём пространства составляет 36^2 или 1296 различных комбинаций и т.д. Следовательно, общий объём пространства входных сообщений (паролей) составит:

36^1 + 36^2 + 36^3 + 36^4 + 36^5 + 36^6 = 2.238.976.116 комбинаций.

Объём радужной таблицы задаётся двумя параметрами: длиной цепочки ключей и количеством этих цепочек в таблице. Если мы выберем длину цепочки равной 2400, а количество цепочек равным 1.000.000, то объём радужной таблицы составит 2.400.000.000 ключей (или вариантов перебора). Это значение превышает общий объём пространства входных сообщений, что нам и требуется.

Используя определённые нами параметры запускаем процесс формирования радужной таблицы, используя команду «**rtgen.exe sha1 loweralpha-numeric 1 6 0 2400 1000000 laba**»:

```
C:\rainbowcrack-1.2-win>rtgen.exe sha1 loweralpha-numeric 1 6 0 2400 1000000
laba
hash routine: sha1
hash length: 20
plain charset: abcdefghijklmnopqrstuvwxyz0123456789
plain charset in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73
7
4 75 76 77 78 79 7a 30 31 32 33 34 35 36 37 38 39
plain length range: 1 - 6
plain charset name: loweralpha-numeric
plain space total: 2238976116
rainbow table index: 0
reduce offset: 0

generating...
100000 of 1000000 rainbow chains generated (4 m 15 s)
200000 of 1000000 rainbow chains generated (4 m 17 s)
300000 of 1000000 rainbow chains generated (4 m 15 s)
400000 of 1000000 rainbow chains generated (4 m 16 s)
500000 of 1000000 rainbow chains generated (4 m 16 s)
600000 of 1000000 rainbow chains generated (4 m 16 s)
700000 of 1000000 rainbow chains generated (4 m 15 s)
800000 of 1000000 rainbow chains generated (4 m 14 s)
900000 of 1000000 rainbow chains generated (4 m 14 s)
1000000 of 1000000 rainbow chains generated (4 m 15 s)
```

Подождав более 50 мин. мы получим необходимый файл таблицы, имеющий название «**sha1_loweralpha-numeric#1-6_0_2400x1000000_laba.rt**» Так как в таблице сохраняются только значения начала и конца цепочек (по 8 байт), то она занимает на диске относительно небольшое место (16.000.000 байт).

Для данной таблицы мы указали значение индекса равным «0». Если мы хотим создать дополнительные таблицы (для увеличения вероятности успешного восстановления пароля), то каждой новой таблице указываем увеличенное на 1 значение индекса.

Следующей процедурой является сортировка таблицы при помощи команды «**rtsort.exe**»:

```
C:\rainbowcrack-1.2-win>rtsort                          sha1_loweralpha-numeric#1-
6_0_2400x1000000_laba.rt
available physical memory: 2147483647 bytes
loading rainbow table...
sorting rainbow table...
writing sorted rainbow table...
C:\rainbowcrack-1.2-win>
```

Создав полностью готовую радужную таблицу мы можем приступать к процессу восстановления паролей пользователей. Для этого подготовим текстовый файл, содержащий исследуемые значения хэш-функции (например, при помощи «Блокнота»). Ниже приведено содержимое такого файла, хранящего в себе 20 значений хэш-функции SHA1.

```
51E7576A524051D8401D8A1EF12BD65D518B55C7
AE0E3AACF507607DEFDAC37574964384FC18F3AC
D9EB49F41C2363970F99584A697785FDB1A4F7B3
5DD38D3D1C516721B79599B4537BFFB56E97C8B0
655DCAA997C5F5DB69D807971DBBF27A2D46E327
18FD7F9D65029F4E7277EE719F2EF382945BFAEB
8D90FFD60B4C905B72EEFD591C23ADC4577BFF3B
4CF9DFB60A62650CAFC36D1D4EC8B147504A9A54
FE2DB16EBABEDCD16605F1677287E84CBB306146
30080914028E006F45AFAD74A2C524AC9BF80F9F
459cd8a3146de2e46635f24b841a5f25045b50cc
7566becb5f181316fe838cd4c301f59d50ff8ecb
b9d6b72964cea634b2b8c44f7a2fedcaed134025
c37e7c8fd457682199909708cf2da1144178e028
290516b50b2382629f73e1f7497e8cfe1cff0b6a
b82ba51fb1bb6528cd011c92aa79612253c0e387
dac63fcb6cd1322ce7aebaaf4016304b141d7a17
257cd5b35797adfc9ecd073a8512e6c66d998a5d
089af482cde49267862d8b515a1a0655bee1d7ea
bf1451bba3bad13f38fedc18a31487a8aff1ceb2
```

В одной строке указывается только одна хэш-функция, которая записывается в виде последовательности шестнадцатеричхных цифр. Другие символы не допустимы.

Допустим мы назвали файл с хэшами как «passw_sha1.txt». Используя команду «**rcrack.exe**», а именно «**`rcrack sha1_loweralpha-numeric#1-6_0_2400x1000000_laba.rt -l passw_sha1.txt`**» осуществляем попытку восстановления паролей пользователей:

```
C:\rainbowcrack-1.2-win>rcrack                          sha1_loweralpha-numeric#1-
6_0_2400x1000000_laba.rt -l passw_sha1.txt
sha1_loweralpha-numeric#1-6_0_2400x1000000_laba.rt:
16000000 bytes read, disk access time: 0.03 s
verifying the file...
searching for 20 hashes...
plaintext of ae0e3aacf507607defdac37574964384fc18f3ac is 9qz2
plaintext of 5dd38d3d1c516721b79599b4537bffb56e97c8b0 is 3ege
plaintext of 18fd7f9d65029f4e7277ee719f2ef382945bfaeb is ukq8
plaintext of 4cf9dfb60a62650cafc36d1d4ec8b147504a9a54 is w7h8
plaintext of fe2db16ebabedcd16605f1677287e84cbb306146 is 7qq7
```

```
plaintext of 30080914028e006f45afad74a2c524ac9bf80f9f is hk7n
plaintext of 459cd8a3146de2e46635f24b841a5f25045b50cc is 9e5yf7
plaintext of 7566becb5f181316fe838cd4c301f59d50ff8ecb is 5e48mw
plaintext of 290516b50b2382629f73e1f7497e8cfe1cff0b6a is x8k2g3
plaintext of b82ba51fb1bb6528cd011c92aa79612253c0e387 is 247qar
plaintext of 257cd5b35797adfc9ecd073a8512e6c66d998a5d is 9thu24
plaintext of 089af482cde49267862d8b515a1a0655bee1d7ea is h64r2x
plaintext of bf1451bba3bad13f38fedc18a31487a8aff1ceb2 is 9f43yv
cryptanalysis time: 50.34 s

statistics
----------------------------------------------------------
plaintext found: 13 of 20 (65.00%)
total disk access time: 0.03 s
total cryptanalysis time: 50.34 s
total chain walk step: 33326659
total false alarm: 14935
total chain walk step due to false alarm: 14218079

result
----------------------------------------------------------
51e7576a524051d8401d8a1ef12bd65d518b55c7 <notfound> hex:<notfound>
ae0e3aacf507607defdac37574964384fc18f3ac 9qz2 hex:39717a32
d9eb49f41c2363970f99584a697785fdb1a4f7b3 <notfound> hex:<notfound>
5dd38d3d1c516721b79599b4537bffb56e97c8b0 3ege hex:33656765
655dcaa997c5f5db69d807971dbbf27a2d46e327 <notfound> hex:<notfound>
18fd7f9d65029f4e7277ee719f2ef382945bfaeb ukq8 hex:756b7138
8d90ffd60b4c905b72eefd591c23adc4577bff3b <notfound> hex:<notfound>
4cf9dfb60a62650cafc36d1d4ec8b147504a9a54 w7h8 hex:77376838
fe2db16ebabedcd16605f1677287e84cbb306146 7qq7 hex:37717137
30080914028e006f45afad74a2c524ac9bf80f9f hk7n hex:686b376e
459cd8a3146de2e46635f24b841a5f25045b50cc 9e5yf7 hex:396535796637
7566becb5f181316fe838cd4c301f59d50ff8ecb 5e48mw hex:356534386d77
b9d6b72964cea634b2b8c44f7a2fedcaed134025 <notfound> hex:<notfound>
c37e7c8fd457682199909708cf2da1144178e028 <notfound> hex:<notfound>
290516b50b2382629f73e1f7497e8cfe1cff0b6a x8k2g3 hex:78386b326733
b82ba51fb1bb6528cd011c92aa79612253c0e387 247qar hex:323437716172
dac63fcb6cd1322ce7aebaaf4016304b141d7a17 <notfound> hex:<notfound>
257cd5b35797adfc9ecd073a8512e6c66d998a5d 9thu24 hex:397468753234
089af482cde49267862d8b515a1a0655bee1d7ea h64r2x hex:683634723278
bf1451bba3bad13f38fedc18a31487a8aff1ceb2 9f43yv hex:396634337976

C:\rainbowcrack-1.2-win>
```

После непродолжительного времени ожидания на экран выводится приведённый выше отчёт. Из него следует, что было восстановлено 13 паролей из 20 (65%) и ушло на это 50,34 секунды. Если бы мы использовали метод полного перебора (метод грубой силы), то у нас на обработку ***каждого*** пароля ушло бы более 50 минут! Учитывая при этом, что пароли сформированы случайным образом (нельзя использовать атаки по словарю и т.д.) и их длина составляет 6 символов.

Всё это показывает эффективность метода «размен время — объём памяти», когда заранее осуществляется большое количество операций по вычислению специальной таблицы, а этап восстановления паролей пользователей затем занимает несколько секунд.

Ниже приведена аналогичная последовательность действий по созданию радужной таблицы

для хэш-функции MD5 и по восстановлению на её основе паролей пользователей:

```
C:\rainbowcrack-1.2-win>rtgen.exe  md5  loweralpha-numeric  1  6  0  2400  1000000
laba

hash routine: md5
hash length: 16
plain charset: abcdefghijklmnopqrstuvwxyz0123456789
plain charset in hex: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73
7
4 75 76 77 78 79 7a 30 31 32 33 34 35 36 37 38 39
plain length range: 1 - 6
plain charset name: loweralpha-numeric
plain space total: 2238976116
rainbow table index: 0
reduce offset: 0

generating...
100000 of 1000000 rainbow chains generated (3 m 48 s)
200000 of 1000000 rainbow chains generated (3 m 48 s)
300000 of 1000000 rainbow chains generated (3 m 47 s)
400000 of 1000000 rainbow chains generated (3 m 48 s)
500000 of 1000000 rainbow chains generated (3 m 47 s)
600000 of 1000000 rainbow chains generated (3 m 50 s)
700000 of 1000000 rainbow chains generated (3 m 46 s)
800000 of 1000000 rainbow chains generated (3 m 46 s)
900000 of 1000000 rainbow chains generated (3 m 46 s)
1000000 of 1000000 rainbow chains generated (3 m 46 s)

C:\rainbowcrack-1.2-win>rtsort                          md5_loweralpha-numeric#1-
6_0_2400x1000000_laba.rt

available physical memory: 2147483647 bytes
loading rainbow table...
sorting rainbow table...
writing sorted rainbow table...

C:\rainbowcrack-1.2-win>rcrack                          md5_loweralpha-numeric#1-
6_0_2400x1000000_laba.rt
-l passw_md5.txt
md5_loweralpha-numeric#1-6_0_2400x1000000_laba.rt:
16000000 bytes read, disk access time: 0.02 s
verifying the file...
searching for 10 hashes...
plaintext of 7e2738461e608bd3cb37c1254abf01e3 is 4q78sw
plaintext of 9b46bc33d4f5488e57809d2e639a74fa is r578fk
plaintext of a171fd24e3b5e58e7dca55adfbfd4a5c is 6dv9y4
plaintext of 225d3dc7ef980c60d5b034befcbf0f6c is 6r8sx4
plaintext of d2ef193311068b8d20bba8b0025f8b10 is tqz235
plaintext of 4b471eee8962c247db66eb6be05bd998 is yq9k72
plaintext of a868905e40cd6bc3108f2744e343ee01 is gp2w78
cryptanalysis time: 20.36 s

statistics
---------------------------------------------------------
plaintext found: 7 of 10 (70.00%)
```

```
total disk access time: 0.02 s
total cryptanalysis time: 20.36 s
total chain walk step: 14869349
total false alarm: 6641
total chain walk step due to false alarm: 6786193

result
------------------------------------------------------
7e2738461e608bd3cb37c1254abf01e3 4q78sw hex:347137387377
9b46bc33d4f5488e57809d2e639a74fa r578fk hex:72353738666b
a171fd24e3b5e58e7dca55adfbfd4a5c 6dv9y4 hex:366476397934
225d3dc7ef980c60d5b034befcbf0f6c 6r8sx4 hex:367238737834
29868768988957f5d34eb3c01e129484 <notfound> hex:<notfound>
d2ef193311068b8d20bba8b0025f8b10 tqz235 hex:74717a323335
69695ae2def03d1b00d10193143fc389 <notfound> hex:<notfound>
d971a9c45c4158b332895940a2b6ee0e <notfound> hex:<notfound>
4b471eee8962c247db66eb6be05bd998 yq9k72 hex:7971396b3732
a868905e40cd6bc3108f2744e343ee01 gp2w78 hex:677032773738

C:\rainbowcrack-1.2-win>
```

Пример варианта задания:

Вариант 1

1. По заданным значениям хэш-функций восстановить пароли пользователей, результаты занести в таблицу. Указать долю правильно восстановленных паролей.
2. Дополнительно сформировать пароли и соответствующие им sha1-хэши, состоящие из русских символов и цифр (длиной не более 6 символов).
3. Сформировать радужную таблицу с использованием символов русского алфавита и набора цифр. Приложить её к отчёту (в электронном виде).
4. Продемонстрировать восстановление паролей, полученных в п.2 используя радужную таблицу, полученную в п.3.

LM-хэш (длина пароля 12 символов):

```
24062F9ABFDF002B79DC190B98842EB7
AAFE7C2888D9722F90044675A76A7AF2
BCC0ABD492B42E84839AF214E797A446
663FE3135116ADD4127BA5C2F43E6AB4
D341284BF612972530D75B03837A8220
```

MD5-хэш (длина пароля 6 символов):

```
2aa1010cc58d6d3fcf6cad27737c8f6e
357b1a263fa0ea406f73f7d16b8532bb
3446a71cc08e28516943567df4676d82
3dc11bbdcb578a8cc28c371430489a48
72cf870a0290c931e39d1c814bbcb0fc
```

SHA1-хэш (длина пароля 6 символов):

```
cbc5446e7dd47a7b82d42e2995edb1df3e5f2db0
6a5912c3781eede62390786fbd43d69c9cca3dc9
aaf044a02f19b365db04873cda6481b175c54141
8a058c61af5dd9d91ffa6007f1499a9154113a34
9af733a383a885f7c07325d71a49c9ae857b53f8
```

Лабораторная работа №6
«ИЗУЧЕНИЕ ПРИНЦИПОВ РАБОТЫ СИММЕТРИЧНОГО КРИПТОАЛГОРИТМА «КУЗНЕЧИК»

Цель работы: исследование принципа работы симметричного блочного криптоалгоритма «Кузнечик» путём редактирования исходного кода криптоалгоритма и его пошаговой отладки.

Порядок выполнения:
1. Ознакомиться с описанием блочного криптоалгоритма «Кузнечик» (Российского стандарта блочного шифрования, www.tk26.ru)
2. Ознакомиться с исходным кодом блочного криптоалгоритма «Кузнечик», в качестве примера дана реализация криптоалгоритма на языке Си от известного криптографа Markku-Juhani O. Saarinen (файлы main.c, kuznechik.h, kuznechik_8bit.c).
3. Скорректировать файл «kuznechik_8bit.c» для добавления команд вывода на экран отладочной информации с результатами выполнения каждой операции криптоалгоритма (операции подстановки, линейных функций L и R, раунда шифрования).
4. Скомпилировать программу командой «gcc -Wall -Ofast -march=native main.c kuznechik_8bit.c -o xtest»
5. Протестировать полученную программу и проанализировать результаты выполнения каждой операции криптоалгоритма (операции подстановки, линейных функций L и R, раунда шифрования).
6. В качестве отчёта по лабораторной работе привести результаты выполнения указанных выше пунктов, в том числе скорректированный исходный код и результаты работы программы.

1. Исходный код файла main.c.

```
// kuznechik.c
// 04-Jan-15  Markku-Juhani O. Saarinen <mjos@iki.fi>

// main() for testing

#include <stdio.h>
#include <time.h>
#include "kuznechik.h"

// debug print state

/*void print_w128(w128_t *x)
{
int i;

for (i = 0; i < 16; i++)
   printf(" %02X", x->b[i]);
printf("\n");
}
*/

// stub main
```

```
void kuz_init();

int main(int argc, char **argv)
{
// These are here in Big Endian format, as that seems to be the favored
// way of representing things. However, it is open if we will have to
// flip byte order for the final version or not.

const uint8_t testvec_key[32] = {
    0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF,
    0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,
    0xFE, 0xDC, 0xBA, 0x98, 0x76, 0x54, 0x32, 0x10,
    0x01, 0x23, 0x45, 0x67, 0x89, 0xAB, 0xCD, 0xEF
};
const uint8_t testvec_pt[16] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
/*const uint8_t testvec_pt[16] = {
    0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x00,
    0xFF, 0xEE, 0xDD, 0xCC, 0xBB, 0xAA, 0x99, 0x88
};*/
const uint8_t testvec_ct[16] = {
    0x7F, 0x67, 0x9D, 0x90, 0xBE, 0xBC, 0x24, 0x30,
    0x5A, 0x46, 0x8D, 0x42, 0xB9, 0xD4, 0xED, 0xCD
};

int i, j, n;
kuz_key_t key;
w128_t x;
uint32_t buf[0x100];
clock_t tim;

printf("Self-test:\n");
kuz_init();

kuz_set_encrypt_key(&key, testvec_key);
for (i = 0; i < 10; i++) {
    printf("K_%d\t=", i + 1);
    print_w128(&key.k[i]);
}

for (i = 0; i < 16; i++)
    x.b[i] = testvec_pt[i];
printf("PT\t=");
print_w128(&x);

kuz_encrypt_block(&key, &x);

printf("CT\t=");
print_w128(&x);

for (i = 0; i < 16; i++) {
    if (testvec_ct[i] != x.b[i]) {
```

```
        fprintf(stderr, "Encryption self-test failure.\n");
        return -1;
    }
}

kuz_set_decrypt_key(&key, testvec_key);
kuz_decrypt_block(&key, &x);

printf("PT\t=");
print_w128(&x);

for (i = 0; i < 16; i++) {
    if (testvec_pt[i] != x.b[i]) {
        fprintf(stderr, "Decryption self-test failure.\n");
        return -2;
    }
}

printf("Self-test OK!\n");

// == Speed Test ==

for (i = 0; i < 0x100; i++)
    buf[i] = i;
kuz_set_encrypt_key(&key, testvec_key);

for (n = 100, tim = 0; tim < 2 * CLOCKS_PER_SEC; n <<= 1) {
    tim = clock();
    for (j = 0; j < n; j++) {
        for (i = 0; i < 0x100; i += 4)
            kuz_encrypt_block(&key, &buf[i]);
    }
    tim = clock() - tim;
    printf("kuz_encrypt_block(): %.3f kB/s (n=%dkB,t=%.3fs)\r",
        ((double) CLOCKS_PER_SEC * n) / ((double) tim),
        n, ((double) tim) / ((double) CLOCKS_PER_SEC));
    fflush(stdout);
}
printf("\n");


for (i = 0; i < 0x100; i++)
buf[i] = i;
kuz_set_decrypt_key(&key, testvec_key);

for (n = 100, tim = 0; tim < 2 * CLOCKS_PER_SEC; n <<= 1) {
    tim = clock();
    for (j = 0; j < n; j++) {
        for (i = 0; i < 0x100; i += 4)
            kuz_decrypt_block(&key, &buf[i]);
    }
    tim = clock() - tim;
    printf("kuz_decrypt_block(): %.3f kB/s (n=%dkB,t=%.3fs)\r",
        ((double) CLOCKS_PER_SEC * n) / ((double) tim),
        n, ((double) tim) / ((double) CLOCKS_PER_SEC));
```

```
    fflush(stdout);
}
printf("\n");


return 0;
}
```

2. Исходный код файла kuznechik_8bit.c.

```
// kuznechik.c
// 04-Jan-15  Markku-Juhani O. Saarinen <mjos@iki.fi>

// This is the basic non-optimized 8-bit "readble" version of the cipher.
// Conforms to included doc/GOSTR-bsh.pdf file, which has an internal
// date of September 2, 2014.

#include <stdio.h>
#include "kuznechik.h"

// The S-Box from section 5.1.1
const uint8_t kuz_pi[0x100] = {
0xFC, 0xEE, 0xDD, 0x11, 0xCF, 0x6E, 0x31, 0x16,  // 00..07
0xFB, 0xC4, 0xFA, 0xDA, 0x23, 0xC5, 0x04, 0x4D,  // 08..0F
0xE9, 0x77, 0xF0, 0xDB, 0x93, 0x2E, 0x99, 0xBA,  // 10..17
0x17, 0x36, 0xF1, 0xBB, 0x14, 0xCD, 0x5F, 0xC1,  // 18..1F
0xF9, 0x18, 0x65, 0x5A, 0xE2, 0x5C, 0xEF, 0x21,  // 20..27
0x81, 0x1C, 0x3C, 0x42, 0x8B, 0x01, 0x8E, 0x4F,  // 28..2F
0x05, 0x84, 0x02, 0xAE, 0xE3, 0x6A, 0x8F, 0xA0,  // 30..37
0x06, 0x0B, 0xED, 0x98, 0x7F, 0xD4, 0xD3, 0x1F,  // 38..3F
0xEB, 0x34, 0x2C, 0x51, 0xEA, 0xC8, 0x48, 0xAB,  // 40..47
0xF2, 0x2A, 0x68, 0xA2, 0xFD, 0x3A, 0xCE, 0xCC,  // 48..4F
0xB5, 0x70, 0x0E, 0x56, 0x08, 0x0C, 0x76, 0x12,  // 50..57
0xBF, 0x72, 0x13, 0x47, 0x9C, 0xB7, 0x5D, 0x87,  // 58..5F
0x15, 0xA1, 0x96, 0x29, 0x10, 0x7B, 0x9A, 0xC7,  // 60..67
0xF3, 0x91, 0x78, 0x6F, 0x9D, 0x9E, 0xB2, 0xB1,  // 68..6F
0x32, 0x75, 0x19, 0x3D, 0xFF, 0x35, 0x8A, 0x7E,  // 70..77
0x6D, 0x54, 0xC6, 0x80, 0xC3, 0xBD, 0x0D, 0x57,  // 78..7F
0xDF, 0xF5, 0x24, 0xA9, 0x3E, 0xA8, 0x43, 0xC9,  // 80..87
0xD7, 0x79, 0xD6, 0xF6, 0x7C, 0x22, 0xB9, 0x03,  // 88..8F
0xE0, 0x0F, 0xEC, 0xDE, 0x7A, 0x94, 0xB0, 0xBC,  // 90..97
0xDC, 0xE8, 0x28, 0x50, 0x4E, 0x33, 0x0A, 0x4A,  // 98..9F
0xA7, 0x97, 0x60, 0x73, 0x1E, 0x00, 0x62, 0x44,  // A0..A7
0x1A, 0xB8, 0x38, 0x82, 0x64, 0x9F, 0x26, 0x41,  // A8..AF
0xAD, 0x45, 0x46, 0x92, 0x27, 0x5E, 0x55, 0x2F,  // B0..B7
0x8C, 0xA3, 0xA5, 0x7D, 0x69, 0xD5, 0x95, 0x3B,  // B8..BF
0x07, 0x58, 0xB3, 0x40, 0x86, 0xAC, 0x1D, 0xF7,  // C0..C7
0x30, 0x37, 0x6B, 0xE4, 0x88, 0xD9, 0xE7, 0x89,  // C8..CF
0xE1, 0x1B, 0x83, 0x49, 0x4C, 0x3F, 0xF8, 0xFE,  // D0..D7
0x8D, 0x53, 0xAA, 0x90, 0xCA, 0xD8, 0x85, 0x61,  // D8..DF
0x20, 0x71, 0x67, 0xA4, 0x2D, 0x2B, 0x09, 0x5B,  // E0..E7
0xCB, 0x9B, 0x25, 0xD0, 0xBE, 0xE5, 0x6C, 0x52,  // E8..EF
0x59, 0xA6, 0x74, 0xD2, 0xE6, 0xF4, 0xB4, 0xC0,  // F0..F7
```

```
0xD1, 0x66, 0xAF, 0xC2, 0x39, 0x4B, 0x63, 0xB6,  // F8..FF
};

// Inverse S-Box
static const uint8_t kuz_pi_inv[0x100] = {
0xA5, 0x2D, 0x32, 0x8F, 0x0E, 0x30, 0x38, 0xC0,  // 00..07
0x54, 0xE6, 0x9E, 0x39, 0x55, 0x7E, 0x52, 0x91,  // 08..0F
0x64, 0x03, 0x57, 0x5A, 0x1C, 0x60, 0x07, 0x18,  // 10..17
0x21, 0x72, 0xA8, 0xD1, 0x29, 0xC6, 0xA4, 0x3F,  // 18..1F
0xE0, 0x27, 0x8D, 0x0C, 0x82, 0xEA, 0xAE, 0xB4,  // 20..27
0x9A, 0x63, 0x49, 0xE5, 0x42, 0xE4, 0x15, 0xB7,  // 28..2F
0xC8, 0x06, 0x70, 0x9D, 0x41, 0x75, 0x19, 0xC9,  // 30..37
0xAA, 0xFC, 0x4D, 0xBF, 0x2A, 0x73, 0x84, 0xD5,  // 38..3F
0xC3, 0xAF, 0x2B, 0x86, 0xA7, 0xB1, 0xB2, 0x5B,  // 40..47
0x46, 0xD3, 0x9F, 0xFD, 0xD4, 0x0F, 0x9C, 0x2F,  // 48..4F
0x9B, 0x43, 0xEF, 0xD9, 0x79, 0xB6, 0x53, 0x7F,  // 50..57
0xC1, 0xF0, 0x23, 0xE7, 0x25, 0x5E, 0xB5, 0x1E,  // 58..5F
0xA2, 0xDF, 0xA6, 0xFE, 0xAC, 0x22, 0xF9, 0xE2,  // 60..67
0x4A, 0xBC, 0x35, 0xCA, 0xEE, 0x78, 0x05, 0x6B,  // 68..6F
0x51, 0xE1, 0x59, 0xA3, 0xF2, 0x71, 0x56, 0x11,  // 70..77
0x6A, 0x89, 0x94, 0x65, 0x8C, 0xBB, 0x77, 0x3C,  // 78..7F
0x7B, 0x28, 0xAB, 0xD2, 0x31, 0xDE, 0xC4, 0x5F,  // 80..87
0xCC, 0xCF, 0x76, 0x2C, 0xB8, 0xD8, 0x2E, 0x36,  // 88..8F
0xDB, 0x69, 0xB3, 0x14, 0x95, 0xBE, 0x62, 0xA1,  // 90..97
0x3B, 0x16, 0x66, 0xE9, 0x5C, 0x6C, 0x6D, 0xAD,  // 98..9F
0x37, 0x61, 0x4B, 0xB9, 0xE3, 0xBA, 0xF1, 0xA0,  // A0..A7
0x85, 0x83, 0xDA, 0x47, 0xC5, 0xB0, 0x33, 0xFA,  // A8..AF
0x96, 0x6F, 0x6E, 0xC2, 0xF6, 0x50, 0xFF, 0x5D,  // B0..B7
0xA9, 0x8E, 0x17, 0x1B, 0x97, 0x7D, 0xEC, 0x58,  // B8..BF
0xF7, 0x1F, 0xFB, 0x7C, 0x09, 0x0D, 0x7A, 0x67,  // C0..C7
0x45, 0x87, 0xDC, 0xE8, 0x4F, 0x1D, 0x4E, 0x04,  // C8..CF
0xEB, 0xF8, 0xF3, 0x3E, 0x3D, 0xBD, 0x8A, 0x88,  // D0..D7
0xDD, 0xCD, 0x0B, 0x13, 0x98, 0x02, 0x93, 0x80,  // D8..DF
0x90, 0xD0, 0x24, 0x34, 0xCB, 0xED, 0xF4, 0xCE,  // E0..E7
0x99, 0x10, 0x44, 0x40, 0x92, 0x3A, 0x01, 0x26,  // E8..EF
0x12, 0x1A, 0x48, 0x68, 0xF5, 0x81, 0x8B, 0xC7,  // F0..F7
0xD6, 0x20, 0x0A, 0x08, 0x00, 0x4C, 0xD7, 0x74      // F8..FF
};

// Linear vector from sect 5.1.2
static const uint8_t kuz_lvec[16] = {
0x94, 0x20, 0x85, 0x10, 0xC2, 0xC0, 0x01, 0xFB,
0x01, 0xC0, 0xC2, 0x10, 0x85, 0x20, 0x94, 0x01
};

// poly multiplication mod p(x) = x^8 + x^7 + x^6 + x + 1
// totally not constant time
static uint8_t kuz_mul_gf256(uint8_t x, uint8_t y)
{
uint8_t z;

z = 0;
while (y) {
    if (y & 1)
        z ^= x;
```

```
    x = (x << 1) ^ (x & 0x80 ? 0xC3 : 0x00);
    y >>= 1;
}
return z;
}

// linear operation l
static void kuz_l(w128_t *w)
{   int i, j;
uint8_t x;

// 16 rounds
for (j = 0; j < 16; j++) {
    // An LFSR with 16 elements from GF(2^8)
    x = w->b[15];  // since lvec[15] = 1

    for (i = 14; i >= 0; i--) {
        w->b[i + 1] = w->b[i];
        x ^= kuz_mul_gf256(w->b[i], kuz_lvec[i]);
    }
    w->b[0] = x;
        /*lab debug*/ printf("       L (%02d)", j);
        /*lab debug*/ print_w128( (w128_t *) w );
}
}

// inverse of linear operation l
static void kuz_l_inv(w128_t *w)
{int i, j;
uint8_t x;

// 16 rounds
for (j = 0; j < 16; j++) {

    x = w->b[0];
    for (i = 0; i < 15; i++) {
        w->b[i] = w->b[i + 1];
        x ^= kuz_mul_gf256(w->b[i], kuz_lvec[i]);
    }
    w->b[15] = x;
}
}

// a no-op
void kuz_init()
{
;
}
// key setup

void kuz_set_encrypt_key(kuz_key_t *kuz, const uint8_t key[32])
{
int i, j;
w128_t c, x, y, z;
```

```
for (i = 0; i < 16; i++) {
    // this will be have to changed for little-endian systems
    x.b[i] = key[i];
    y.b[i] = key[i + 16];
}

kuz->k[0].q[0] = x.q[0];
kuz->k[0].q[1] = x.q[1];
kuz->k[1].q[0] = y.q[0];
kuz->k[1].q[1] = y.q[1];

for (i = 1; i <= 32; i++) {
    // C Value
    c.q[0] = 0;
    c.q[1] = 0;
    c.b[15] = i;       // load round in lsb
    kuz_l(&c);

    z.q[0] = x.q[0] ^ c.q[0];
    z.q[1] = x.q[1] ^ c.q[1];
    for (j = 0; j < 16; j++)
        z.b[j] = kuz_pi[z.b[j]];
    kuz_l(&z);
    z.q[0] ^= y.q[0];
    z.q[1] ^= y.q[1];

    y.q[0] = x.q[0];
    y.q[1] = x.q[1];

    x.q[0] = z.q[0];
    x.q[1] = z.q[1];

    if ((i & 7) == 0) {
        kuz->k[(i >> 2)].q[0] = x.q[0];
        kuz->k[(i >> 2)].q[1] = x.q[1];
        kuz->k[(i >> 2) + 1].q[0] = y.q[0];
        kuz->k[(i >> 2) + 1].q[1] = y.q[1];
    }
}
}

// these two funcs are identical in the simple implementation
void kuz_set_decrypt_key(kuz_key_t *kuz, const uint8_t key[32])
{
kuz_set_encrypt_key(kuz, key);
}

// encrypt a block - 8 bit way
void kuz_encrypt_block(kuz_key_t *key, void *blk)
{int i, j;
w128_t x;

x.q[0] = ((uint64_t *) blk)[0];
x.q[1] = ((uint64_t *) blk)[1];
       /*lab debug*/ printf("ENCRYPT \n");
```

```
        /*lab debug*/ printf("PLAIN TEXT   ");
        /*lab debug*/ print_w128( (w128_t *)blk );
for (i = 0; i < 9; i++) {
    x.q[0] ^= key->k[i].q[0];
    x.q[1] ^= key->k[i].q[1];
        /*lab debug*/ printf("ROUND KEY (%d)", i);
        /*lab debug*/ print_w128( (w128_t *) &key->k[i] );

        /*lab debug*/ printf("    result X ");
        /*lab debug*/ print_w128( (w128_t *) &x );
    for (j = 0; j < 16; j++)
        x.b[j] = kuz_pi[x.b[j]];

        /*lab debug*/ printf("    result PI");
        /*lab debug*/ print_w128( (w128_t *) &x );
    kuz_l(&x);
        /*lab debug*/ printf("    result L ");
        /*lab debug*/ print_w128( (w128_t *) &x );

}
((uint64_t *) blk)[0] = x.q[0] ^ key->k[9].q[0];
((uint64_t *) blk)[1] = x.q[1] ^ key->k[9].q[1];

        /*lab debug*/ printf("CYPHER TEXT  ");
        /*lab debug*/ print_w128( (w128_t *) &x );
        /*lab debug*/ printf("\n");
}

// decrypt a block - 8 bit way
void kuz_decrypt_block(kuz_key_t *key, void *blk)
{
int i, j;
w128_t x;

x.q[0] = ((uint64_t *) blk)[0] ^ key->k[9].q[0];
x.q[1] = ((uint64_t *) blk)[1] ^ key->k[9].q[1];

for (i = 8; i >= 0; i--) {

    kuz_l_inv(&x);
    for (j = 0; j < 16; j++)
        x.b[j] = kuz_pi_inv[x.b[j]];

    x.q[0] ^= key->k[i].q[0];
    x.q[1] ^= key->k[i].q[1];
}
((uint64_t *) blk)[0] = x.q[0];
((uint64_t *) blk)[1] = x.q[1];
}

// debug print state
void print_w128(w128_t *x)
{int i;

for (i = 0; i < 16; i++)
```

```
    printf(" %02X", x->b[i]);
printf("\n");
}
```

Команда для создания исполняемого файла:

gcc -Wall -Ofast -march=native main.c kuznechik_8bit.c -o xtest

Результат выполнения программы:
/laba_crypt/kuznechik-master$ ./xtest

```
K_1 =  88 99 AA BB CC DD EE FF 00 11 22 33 44 55 66 77
K_2 = FE DC BA 98 76 54 32 10 01 23 45 67 89 AB CD EF
K_3 = DB 31 48 53 15 69 43 43 22 8D 6A EF 8C C7 8C 44
K_4 = 3D 45 53 D8 E9 CF EC 68 15 EB AD C4 0A 9F FD 04
K_5 = 57 64 64 68 C4 4A 5E 28 D3 E5 92 46 F4 29 F1 AC
K_6 = BD 07 94 35 16 5C 64 32 B5 32 E8 28 34 DA 58 1B
K_7 = 51 E6 40 75 7E 87 45 DE 70 57 27 26 5A 00 98 B1
K_8 = 5A 79 25 01 7B 9F DD 3E D7 2A 91 A2 22 86 F9 84
K_9 = BB 44 E2 53 78 C7 31 23 A5 F3 2F 73 CD B6 E5 17
K_10   = 72 E9 DD 74 16 BC F4 5B 75 5D BA A8 8E 4A 40 43
PT  = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ENCRYPT
PLAIN TEXT     00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ROUND KEY (0) 88 99 AA BB CC DD EE FF 00 11 22 33 44 55 66 77
    result X  88 99 AA BB CC DD EE FF 00 11 22 33 44 55 66 77
    result PI D7 E8 38 7D 88 D8 6C B6 FC 77 65 AE EA 0C 9A 7E
       L (00) BC D7 E8 38 7D 88 D8 6C B6 FC 77 65 AE EA 0C 9A
       L (01) 50 BC D7 E8 38 7D 88 D8 6C B6 FC 77 65 AE EA 0C
       L (02) 6F 50 BC D7 E8 38 7D 88 D8 6C B6 FC 77 65 AE EA
       L (03) 08 6F 50 BC D7 E8 38 7D 88 D8 6C B6 FC 77 65 AE
       L (04) E7 08 6F 50 BC D7 E8 38 7D 88 D8 6C B6 FC 77 65
       L (05) B3 E7 08 6F 50 BC D7 E8 38 7D 88 D8 6C B6 FC 77
       L (06) 8F B3 E7 08 6F 50 BC D7 E8 38 7D 88 D8 6C B6 FC
       L (07) 92 8F B3 E7 08 6F 50 BC D7 E8 38 7D 88 D8 6C B6
       L (08) FF 92 8F B3 E7 08 6F 50 BC D7 E8 38 7D 88 D8 6C
       L (09) E9 FF 92 8F B3 E7 08 6F 50 BC D7 E8 38 7D 88 D8
       L (10) 85 E9 FF 92 8F B3 E7 08 6F 50 BC D7 E8 38 7D 88
       L (11) CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC D7 E8 38 7D
       L (12) 2E CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC D7 E8 38
       L (13) 90 2E CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC D7 E8
       L (14) E3 90 2E CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC D7
       L (15) 57 E3 90 2E CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC
    result L  57 E3 90 2E CA 85 E9 FF 92 8F B3 E7 08 6F 50 BC
….

ROUND KEY (8) BB 44 E2 53 78 C7 31 23 A5 F3 2F 73 CD B6 E5 17
    result X  A1 51 67 65 EA 13 12 82 E0 F1 8A 38 A7 FD EE 5C
    result PI 97 70 C7 7B 25 DB F0 24 20 A6 D6 06 44 4B 6C 9C
       L (00) E3 97 70 C7 7B 25 DB F0 24 20 A6 D6 06 44 4B 6C
       L (01) CE E3 97 70 C7 7B 25 DB F0 24 20 A6 D6 06 44 4B
       L (02) 40 CE E3 97 70 C7 7B 25 DB F0 24 20 A6 D6 06 44
```

```
       L (03) C2 40 CE E3 97 70 C7 7B 25 DB F0 24 20 A6 D6 06
       L (04) 31 C2 40 CE E3 97 70 C7 7B 25 DB F0 24 20 A6 D6
       L (05) 91 31 C2 40 CE E3 97 70 C7 7B 25 DB F0 24 20 A6
       L (06) AD 91 31 C2 40 CE E3 97 70 C7 7B 25 DB F0 24 20
       L (07) 73 AD 91 31 C2 40 CE E3 97 70 C7 7B 25 DB F0 24
       L (08) BE 73 AD 91 31 C2 40 CE E3 97 70 C7 7B 25 DB F0
       L (09) 05 BE 73 AD 91 31 C2 40 CE E3 97 70 C7 7B 25 DB
       L (10) 20 05 BE 73 AD 91 31 C2 40 CE E3 97 70 C7 7B 25
       L (11) 30 20 05 BE 73 AD 91 31 C2 40 CE E3 97 70 C7 7B
       L (12) 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3 97 70 C7
       L (13) 1C 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3 97 70
       L (14) 57 1C 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3 97
       L (15) E6 57 1C 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3
    result L  E6 57 1C 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3
CYPHER TEXT   E6 57 1C 2A 30 20 05 BE 73 AD 91 31 C2 40 CE E3

CT  = 94 BE C1 5E 26 9C F1 E5 06 F0 2B 99 4C 0A 8E A0
```

Лабораторная работа №7
«ИССЛЕДОВАНИЕ УЯЗВИМОСТИ «ПЕРЕПОЛНЕНИЕ БУФЕРА»

Цель работы: исследование уязвимости «переполнение буфера» на примере тестовой программы и операционной системы Ubuntu/Linux.

Порядок выполнения:

1. Скорректировать тестовый файл «main.c» в соответствии с вариантом задания.
2. Скомпилировать программу командой «gcc -O0 -mpreferred-stack-boundary=2 -g -m32 -fno-stack-protector main_var?.c»
3. Протестировать полученную программу путём ввода неправильного пароля, правильного пароля и неправильного пароля, длина которого превышает размер выделенного буфера.
4. Запустить отладчик **gdb**, задать точку останова на функцию **main()** и получить результат дизассемблирования функции **main()** и **buff_overflow_test()**.
5. При помощи пошаговой отладки программы в **gdb** посмотреть содержимое стека для буфера **buff_var** для случаев произвольного короткого пароля и пароля, длина которого превышает размер буфера на 1 байт. Привести результат выполнения программы при вводе этих паролей.
6. В качестве отчёта по лабораторной работе привести результаты выполнения указанных выше пунктов, в том числе скорректированный исходный код и результаты работы отладчика **gdb**.

Таблица 1 – Варианты заданий

№ бригады	**1**	**2**	**3**	**4**	**5**
Название исходного файла	main_var1	main_var2	main_var3	main_var4	main_var5
Название буфера	buff_var1	buff_var2	buff_var3	buff_var4	buff_var5
Размер буфера, байт	5	6	7	8	9
Правильный пароль	pass1	pass2	pass3	pass4	pass5

Таблица 1 – Варианты заданий (продолжение)

№ бригады	**6**	**7**	**8**	**9**	**10**
Название исходного файла	main_var6	main_var7	main_var8	main_var9	main_var10
Название буфера	buff_var6	buff_var7	buff_var8	buff_var9	buff_var10
Размер буфера, байт	10	11	12	13	14
Правильный пароль	pass6	pass7	pass8	pass9	pass10

Пример выполнения

Необходимое программное обеспечение:

1. Операционная система Ubuntu/Linux.
2. Установленные компилятор **gcc** и отладчик **gdb**.
3. Редактор текста с подсветкой синтаксиса **gedit** (или другой).

Вариант для примера:

- название исходного файла – main.c
- название буфера – buff_var0
- размер буфера, байт – 4
- правильный пароль – pasw

1. Исходный код тестовой программы (main.c).

```
#include <stdio.h>
#include <string.h>

int main(void)
{
  buff_overflow_test();
  return 0;
}

int buff_overflow_test()
{
  char buff_var0[4];                ; исследуемый буфер
  int pass = 0;

  printf("\n Enter the password : \n");
  gets(buff_var0);                  ; ввод значений в буфер

  if(strcmp(buff_var0, "pasw"))     ; сравнение значения
  {
    printf ("\n Wrong Password \n");
  }
  else
  {
    printf ("\n Correct Password \n");
    pass = 1;
  }

  if(pass)                  ; участок кода, выполняемый
  {                         ; если пароль верный
   /* Now Give root or admin rights to user*/
    printf ("\n Root privileges given to the user \n");
  }
}
```

Команда для создания исполняемого файла с отключением механизмов защиты адресного пространства и оптимизаций:

gcc -O0 -mpreferred-stack-boundary=2 -g -m32 -fno-stack-protector main.c

Результат выполнения компилятора:

```
main.c: In function ‘main’:
main.c:7:5: warning: implicit declaration of function ‘buff_overflow_test’ [-Wimplicit-function-declaration]
    buff_overflow_test();
    ^
main.c: In function ‘buff_overflow_test’:
main.c:19:5: warning: implicit declaration of function ‘gets’ [-Wimplicit-function-declaration]
    gets(buff_var0);
    ^
/tmp/ccE51AAQ.o: In function `buff_overflow_test':
/laba_buff_overflow/src_lab/main.c:19: warning: the `gets' function is dangerous and should not be used.
```

2. Тестирование программы.

Запускаем исполняемый файл командой «./a.out» и вводим неверный пароль «pass»:

```
 Enter the password :
pass

 Wrong Password
```

Как видно, в случае ввода неправильного пароля программа отработала правильно и вывела сообщение о неправильном пароле.

Запускаем исполняемый файл командой «./a.out» и вводим верный пароль «pasw»:

```
 Enter the password :
pasw

 Correct Password

 Root privileges given to the user
```

В случае ввода правильного пароля программа отработала правильно и вывела сообщение о правильности пароля и о получении повышенных привилегий.

Запуск исполняемого файла командой «./a.out» и ввод неверного пароля «11111111111», который больше размера буфера:

```
 Enter the password :
11111111111

 Wrong Password

 Root privileges given to the user
```

В данном случае, программа выводит сообщение о том, что пароль неправильный, но были получены повышенные привилегии. Данный эффект достигается за счёт того, что введённые данные вышли за границы отведённого буфера и затёрли соседние области в стеке, в том числе переменную «**int pass**». Так как значение этой переменной отличается от «0», то происходит выполнение участка кода, дающего повышенные привилегии:

```
if(pass)                          ; участок кода, выполняемый
{                              ; если пароль верный
  /* Now Give root or admin rights to user*/
  printf ("\n Root privileges given to the user \n");
}
```

3. Отладка программы.

Запускаем отладчик **gdb**:

```
 laba_buff_overflow/src_lab$ gdb a.out
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...done.
(gdb)
```

Добавляем точку остановки на функцию **main** командой **break main** :

```
(gdb) break main
Breakpoint 1 at 0x804846e: file main.c, line 7.
```

Выводим дизассемблированный код для функции **main** и **buff_overflow_test** командами **disassemble main** и **disassemble buff_overflow_test** :

```
(gdb) disassemble main
Dump of assembler code for function main:
   0x0804846b <+0>:    push   %ebp
   0x0804846c <+1>:    mov    %esp,%ebp
   0x0804846e <+3>:    call   0x804847a <buff_overflow_test>
   0x08048473 <+8>:    mov    $0x0,%eax
   0x08048478 <+13>:   pop    %ebp
   0x08048479 <+14>:   ret
End of assembler dump.
(gdb) disassemble buff_overflow_test
Dump of assembler code for function buff_overflow_test:
   0x0804847a <+0>:    push   %ebp
   0x0804847b <+1>:    mov    %esp,%ebp
   0x0804847d <+3>:    sub    $0x8,%esp      ; выделение памяти в стеке для переменных
   0x08048480 <+6>:    movl   $0x0,-0x4(%ebp)
   0x08048487 <+13>:   push   $0x8048570
   0x0804848c <+18>:   call   0x8048340 <puts@plt> ; вызов функции printf()
   0x08048491 <+23>:   add    $0x4,%esp
   0x08048494 <+26>:   lea    -0x8(%ebp),%eax
   0x08048497 <+29>:   push   %eax
   0x08048498 <+30>:   call   0x8048330 <gets@plt> ; вызов функции gets()
   0x0804849d <+35>:   add    $0x4,%esp
   0x080484a0 <+38>:   push   $0x8048588
   0x080484a5 <+43>:   lea    -0x8(%ebp),%eax
   0x080484a8 <+46>:   push   %eax
   0x080484a9 <+47>:   call   0x8048320 <strcmp@plt> ; сравнение строк
   0x080484ae <+52>:   add    $0x8,%esp
   0x080484b1 <+55>:   test   %eax,%eax
   0x080484b3 <+57>:   je     0x80484c4 <buff_overflow_test+74>
   0x080484b5 <+59>:   push   $0x804858d
   0x080484ba <+64>:   call   0x8048340 <puts@plt>
   0x080484bf <+69>:   add    $0x4,%esp
   0x080484c2 <+72>:   jmp    0x80484d8 <buff_overflow_test+94>
   0x080484c4 <+74>:   push   $0x804859f
   0x080484c9 <+79>:   call   0x8048340 <puts@plt>
   0x080484ce <+84>:   add    $0x4,%esp
   0x080484d1 <+87>:   movl   $0x1,-0x4(%ebp)
   0x080484d8 <+94>:   cmpl   $0x0,-0x4(%ebp)
   0x080484dc <+98>:   je     0x80484eb <buff_overflow_test+113>
   0x080484de <+100>:  push   $0x80485b4
   0x080484e3 <+105>:  call   0x8048340 <puts@plt>
   0x080484e8 <+110>:  add    $0x4,%esp
   0x080484eb <+113>:  nop
   0x080484ec <+114>:  leave
   0x080484ed <+115>:  ret
```

```
End of assembler dump.
(gdb)
```

Обратим внимание на регистры указателей стека: ESP – текущий указатель стека, EBP – базовый указатель стека.

В строках

```
0x0804847a <+0>:    push   %ebp
0x0804847b <+1>:    mov    %esp,%ebp
```

сохраняется значение базового указателя стека и присваивание базовому указателю стека значения текущего указателя стека.

В строке

```
0x0804847d <+3>:    sub    $0x8,
```

показано изменение указателя стека на 8 байт командой **sub $0x8,%esp**. Таким образом в стеке выделяется 8 байт под переменные **buff_var0** и **pass** (4 байта для буфера и 4 байта для целочисленной переменной **pass**).

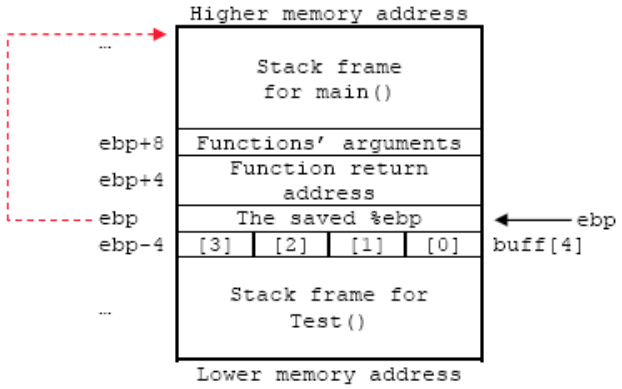


Рис. 1. Схема работы стека

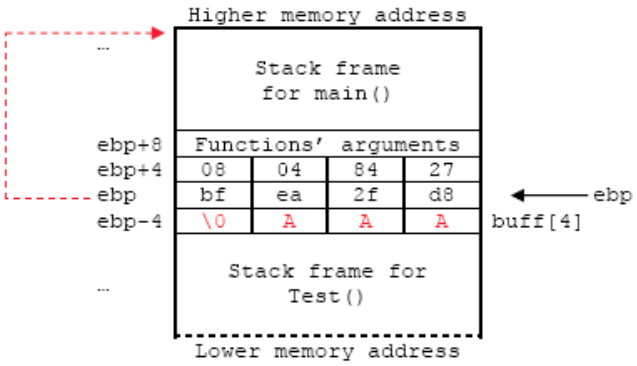


Рис. 2. Схема работы стека при вводе значения пароля «AAA»

Выйдем из отладчика **gdb** нажатием комбинации кнопок «CTRL-Z» и перезапустим его командой **gdb -q a.out**.

Установим точку остановки командой **break main**, после чего запустим программу командой **r** (run) и осуществим пошаговое выполнение программы командой **s** (step):

```
Reading symbols from a.out...done.
```

```
(gdb) break main
Breakpoint 1 at 0x804846e: file main.c, line 7.
(gdb) r
Starting program: /laba_buff_overflow/src_lab/a.out

Breakpoint 1, main () at main.c:7
7       buff_overflow_test();
(gdb) s
buff_overflow_test () at main.c:16
16      int pass = 0;
(gdb)
18      printf("\n Enter the password : \n");
(gdb)

 Enter the password :
19      gets(buff_var0);
```

В поле ввода введём пароль «AAA»

```
(gdb)
AAA
21      if(strcmp(buff_var0, "pasw"))
```

На 21 строке, в которой осуществляется сравнение введённого пароля с «pasw», посмотрим командами **x/x** и **x/s** содержимое памяти в стеке по адресу, указанному в регистре ESP:

```
(gdb) x/x $esp
0xffffcc98:   0x00414141
(gdb) x/s $esp
0xffffcc98:   "AAA"
```

Как видно, по заданному адресу хранится введённый нами пароль.

Посмотрим, что хранится в соседней области памяти. Мы знаем, что на наш буфер выделено 4 байта. Поэтому, прибавив значение 4 к значению указателя сможем посмотреть содержимое переменной «**int pass**»:

```
(gdb) x/s $esp+4
0xffffcc9c:   ""
(gdb) x/x $esp+4
0xffffcc9c:   0x00
(gdb)
```

Как видно, содержимое переменной «**int pass**» равно 0x00.

Перезапустим отладчик и повторим предыдущие шаги, только в качестве пароля введём значение «AAAAA», что на один байт больше, чем выделяется на буфер.

```
(gdb) x/x $esp
0xffffcc98:   0x41414141
(gdb) x/s $esp
```

```
0xffffcc98:   "AAAAA"
(gdb) x/x $esp+4
0xffffcc9c:   0x41
(gdb) x/s $esp+4
0xffffcc9c:   "A"
```

Как видно, содержимое переменной «**int pass**» равно 0x41. Если продолжить выполнение программы, то мы увидим, что пароль неправильный, но предоставлен привилегированный режим.

```
(gdb) s
23          printf ("\n Wrong Password \n");
(gdb)

 Wrong Password
31      if(pass)
(gdb)
34          printf ("\n Root privileges given to the user \n");
(gdb)

 Root privileges given to the user
36  }
(gdb)
main () at main.c:8
8       return 0;
(gdb)
```

Таким образом, успешно продемонстрирована возможность переполнения буфера для манипуляции значениями другой переменной.

Лабораторная работа №8
«НАСТРОЙКА VPN-СОЕДИНЕНИЯ МЕЖДУ МСЭ PFSENSE (OPENVPN-PSK)»

Цель работы: развёртывание двух виртуальных защищённых сегментов сети, объединённых в VPN-сеть, при помощи технологии виртуализации (VirtualBox) и межсетевого экрана PfSense.

Задание:

1. Развернуть и запустить виртуальные машины с межсетевым экраном PfSense и операционной системой Ubuntu.
2. Осуществить настройку межсетевых экранов PfSense.
3. Проверить правильность прохождения сетевых пакетов через защищённое соединение.
4. Зафиксировать результаты настроек и проверок в отчёте.

Порядок выполнения:

1. Задать параметры виртуальной машины, предназначенной под установку межсетевого экрана PfSense.
2. Установить на подготовленную виртуальную машину межсетевой экран PfSense.
3. Через консоль на виртуальной машине определить IP адрес и параметры для получения доступа к web-интерфейсу межсетевого экрана PfSense, при необходимости изменить IP адрес доступа.
4. Запустить на виртуальной машине PC1 образ операционной системы Ubuntu.
5. Задать параметры виртуальной машины PC1 для получения доступа к виртуальному сегменту сети «vnet1».
6. Настроить начальные параметры межсетевого экрана PfSense, для управления использовать виртуальную машину PC1.
7. Настроить параметры VPN-соединения для межсетевого экрана PfSense.
8. Повторить аналогичные шаги 1-7 для виртуального сегмента «vnet2» (делает другая бригада).
9. Проверить прохождение сетевых пакетов через VPN-соединение.
10. В качестве отчёта по лабораторной работе привести содержимое конфигурационных файлов виртуальных машин (расширение *.vbox) и снимки экранов при настройке межсетевого экрана PfSense через web-интерфейс.

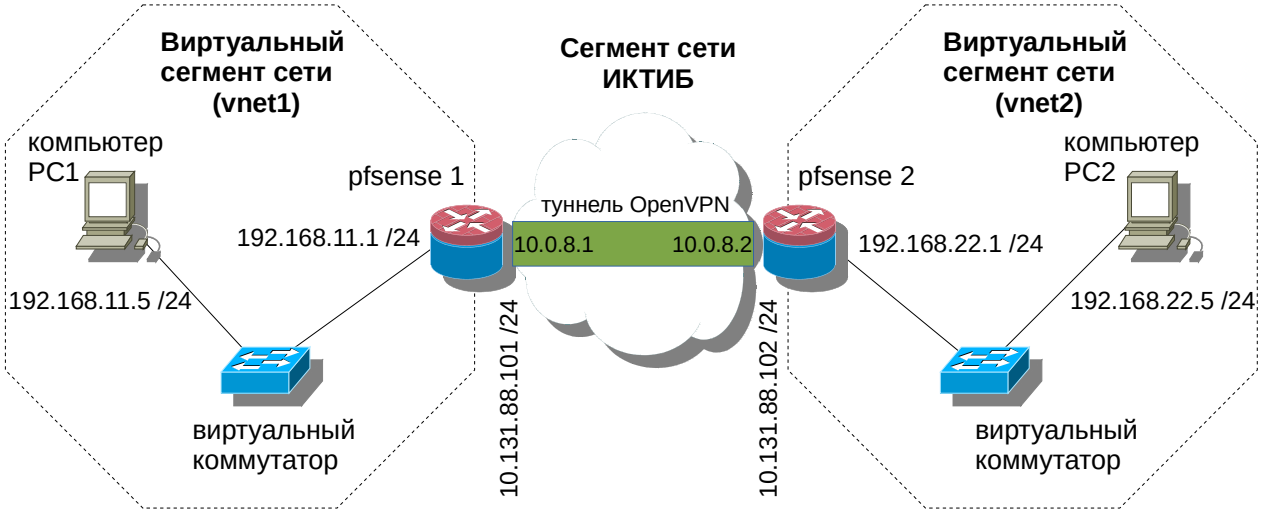
Таблица 1 – Варианты заданий

№ бригады	1	2	3	4	5
Имя межсетевого экрана	pf1	pf2	pf3	pf4	pf5
Имя виртуального сегмента сети	vnet1	vnet2	vnet3	vnet4	vnet5
IP-адрес внешнего интерфейса	10.131.88.101 /24	10.131.88.102 /24	10.131.88.103 /24	10.131.88.104 /24	10.131.88.105 /24
IP-адрес внутреннего интерфейса	192.168.11.1 /24	192.168.22.1 /24	192.168.33.1 /24	192.168.44.1 /24	192.168.55.1 /24
Шлюз для внешнего интерфейса	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24
IP-адрес PC1	192.168.11.5 /24	192.168.22.5 /24	192.168.33.5 /24	192.168.44.5 /24	192.168.55.5 /24

Таблица 1 – Варианты заданий (продолжение)

№ бригады	6	7	8	9	10
Имя межсетевого экрана	pf6	pf7	pf8	pf9	pf10
Имя виртуального сегмента сети	vnet6	vnet7	vnet8	vnet9	vnet101
IP-адрес внешнего интерфейса	10.131.88.106 /24	10.131.88.107 /24	10.131.88.108 /24	10.131.88.109 /24	10.131.88.110 /24
IP-адрес внутреннего интерфейса	192.168.66.1 /24	192.168.77.1 /24	192.168.88.1 /24	192.168.99.1 /24	192.168.101.1 /24
Шлюз для внешнего интерфейса	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24	10.131.88.1/24
IP-адрес PC1	192.168.66.5 /24	192.168.77.5 /24	192.168.88.5 /24	192.168.99.5 /24	192.168.101.5 /24

Пример топологии сети:



Пример выполнения

Необходимое программное обеспечение:

1. Установленная программа виртуализации VirtualBox.
2. ISO-образ межсетевого экрана PfSense.
3. Файлы виртуальной машины «ubuntu_min» для виртуализации компьютеров PC1 и PC2.

1. Развёртывание виртуального сегмента vnet1.

Запускаем программу VirtualBox.

Создаём новую виртуальную машину для межсетевого экрана PfSense:

Создать→Имя «pfsense_1», тип ОС «BSD», версия ОС «FreeBSD».

Объём памяти: 512 МБ

Не подключать виртуальный жёсткий диск.

Подтвердить создание виртуальной машины.

Производим настройку виртуальной машины «pfsense_1»

Во вкладке «Система»:

Отключаем все носители, кроме CD/DVD.

Включаем I/O APIC.

Включаем все возможности по аппаратной виртуализации.

Во вкладке «Носители»:

Указываем для CD/DVD ISO-образ межсетевого экрана PfSense.

Во вкладке «Сеть»:

Включаем «Адаптер 1»:

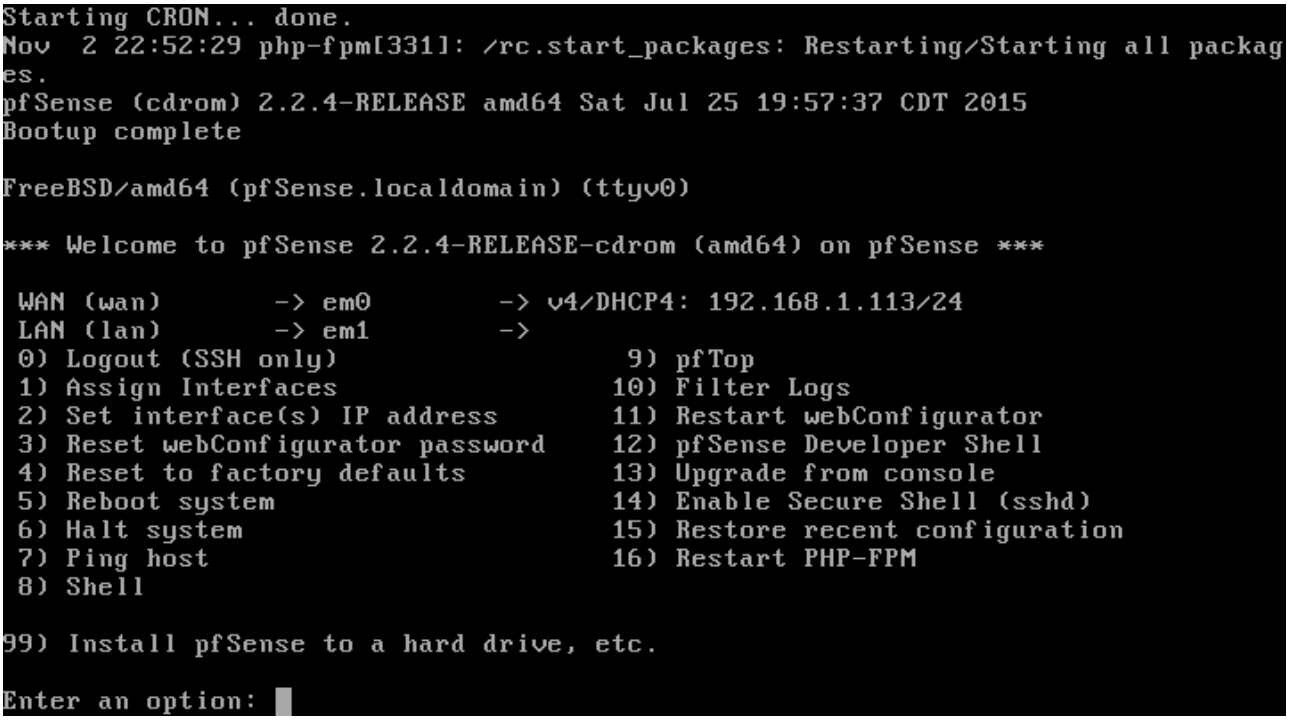
Указываем тип подключения «Сетевой мост», имя интерфейса «eth1» (тот, который подключен к внешней сети. Просмотр интерфейсов можно осуществить через консольную команду ifconfig.)

Данный интерфейс будем использовать как внешний интерфейс межсетевого экрана PfSense.

Включаем «Адаптер 2»:

Указываем тип подключения «Внутренняя сеть», имя «vnet1», кабель подключен.

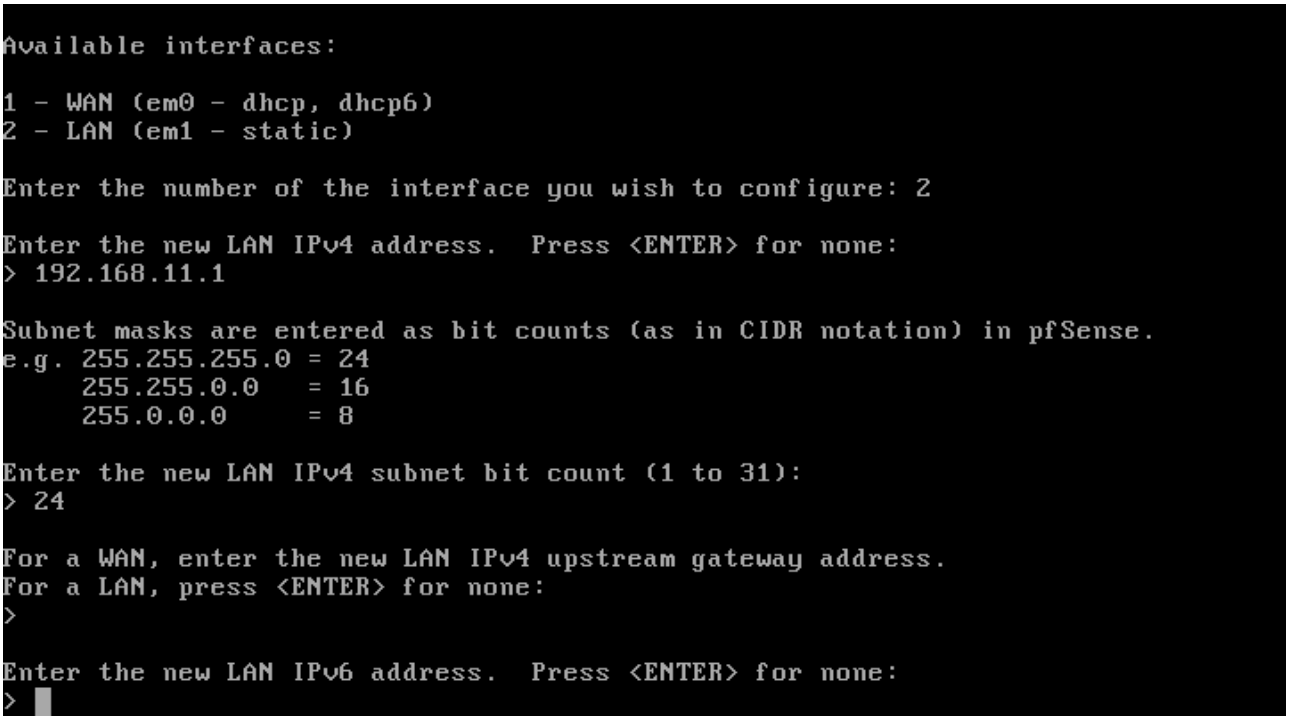
Запускаем виртуальную машину «pfsense_1» и выбираем режим загрузки «Boot Multi User». После успешной загрузки в консоли будет выведена оболочка управления.



При необходимости следует настроить IP адреса для внешнего (em0) и внутреннего интерфейсов (em1).

Настройка IP адреса для внутреннего интерфейса:

Нажать кнопку «2» для активизации настройки IP-адреса, выбрать интерфейс LAN, задать новый IP-адрес и указать для него маску подсети.



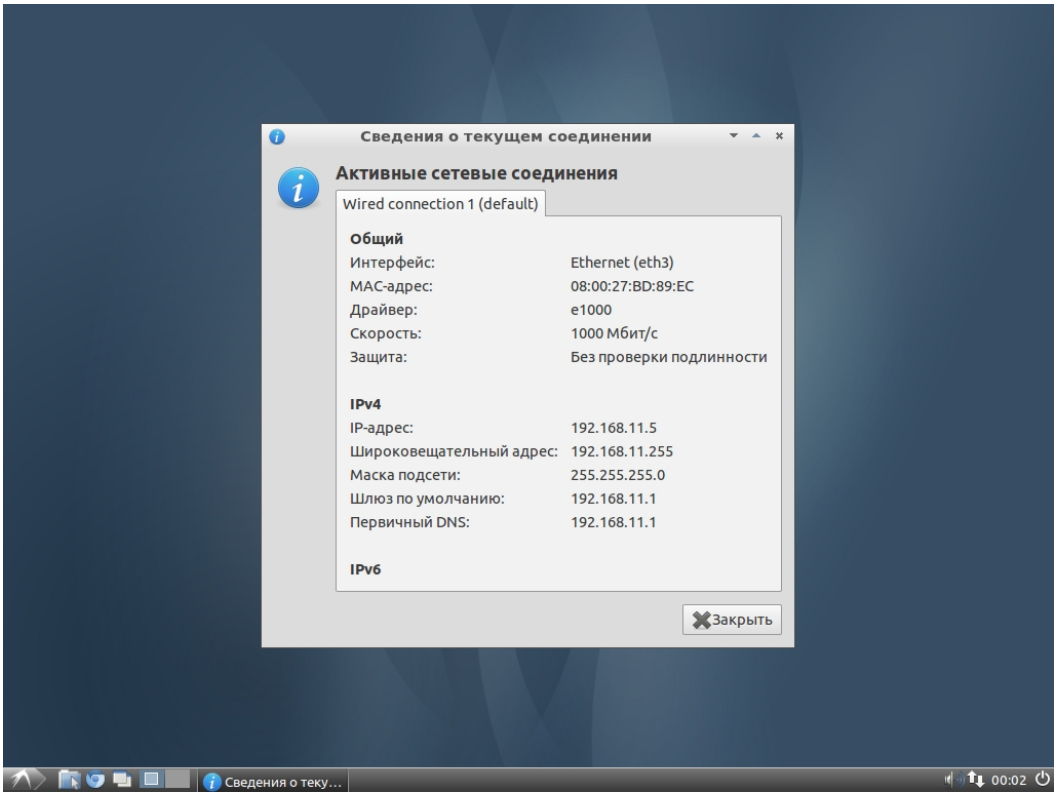
Активировать DHCP сервер на интерфейсе LAN и указать начальный и конечный IP-адреса для автоматического назначения узлам внутренней сети.

```
2 - LAN (em1 - static)

Enter the number of the interface you wish to configure: 2

Enter the new LAN IPv4 address.  Press <ENTER> for none:
> 192.168.11.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address.  Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 192.168.11.5
Enter the end address of the IPv4 client address range: 192.168.11.20
```

Активировать перенастройку webConfigurator под новые параметры интерфейса.

```
For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address.  Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 192.168.11.5
Enter the end address of the IPv4 client address range: 192.168.11.20

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) y

Please wait while the changes are saved to LAN...
 Reloading filter...
 Reloading routing configuration...
 DHCPD...
 Restarting webConfigurator...

The IPv4 LAN address has been set to 192.168.11.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
                http://192.168.11.1/

Press <ENTER> to continue.
```

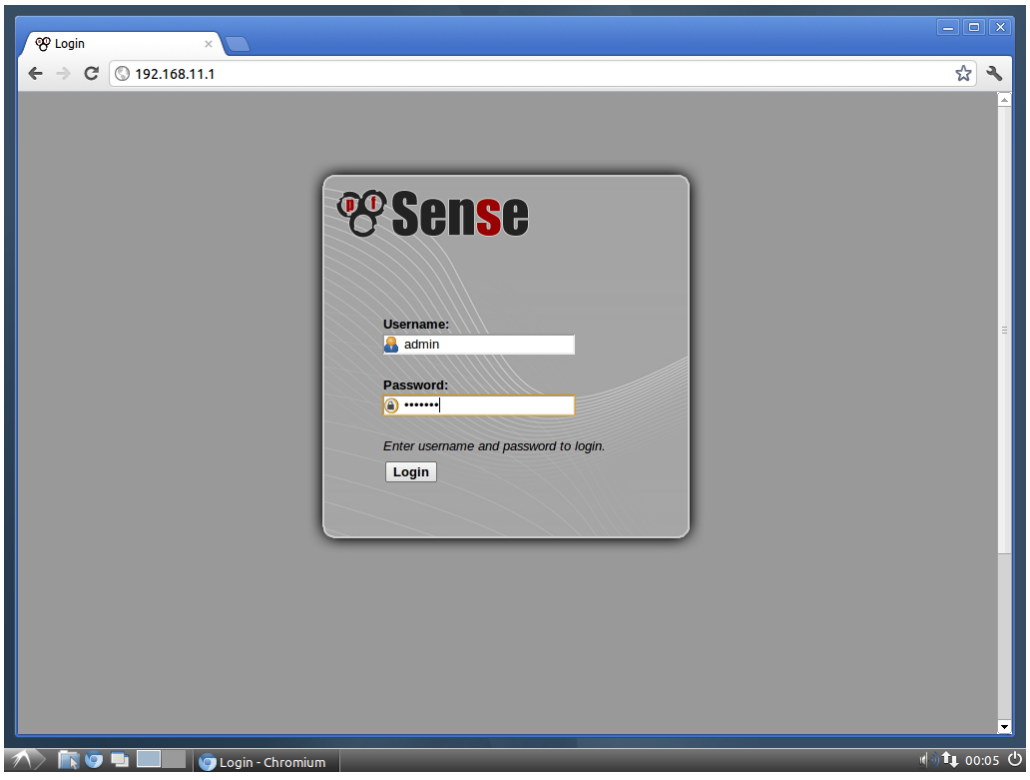
Запускаем виртуальную машину «ubuntu_min», осуществляем вход в ОС используя login: student и passw: studentstudent проверяем сетевые настройки.



Проверяем связь с внутренним интерфейсом виртуальной машины «pfsense_1»:

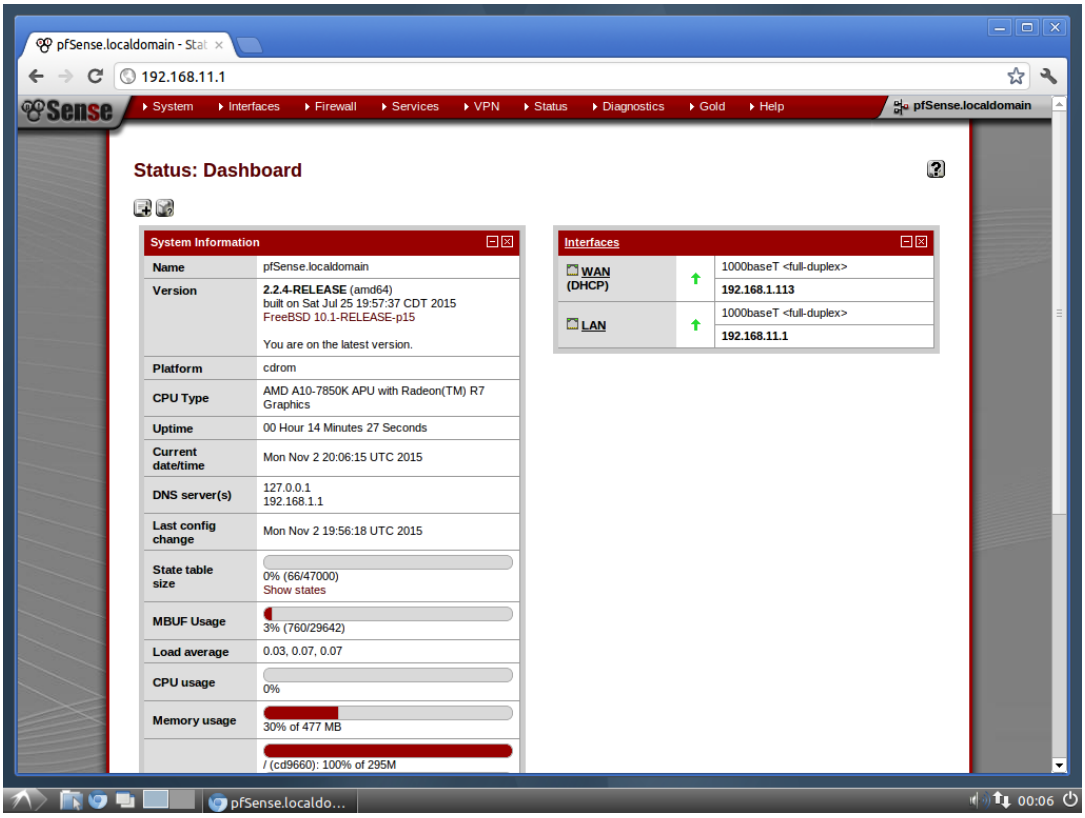
ping 192.168.11.1

Запускаем на «ubuntu_min» веб-браузер (chromium) и пытаемся получить доступ к web-интерфейсу управления «pfsense_1». Для этого в строке адреса вводим IP-адрес внутреннего интерфейса «pfsense_1» (192.168.11.1).



После успешного подключения к web-интерфейсу управления «pfsense_1» указываем в качестве username: admin и Password: pfsense.

Результат успешного входа в web-интерфейс управления «pfsense_1»:



2. Настройка «pfsense_1» через web-интерфейс управления.

Создаём OpenVPN сервер:

VPN → OpenVPN → Add Server

Задаём параметры OpenVPN сервера:

Server Mode → to Peer to Peer (Shared Key) (PSK)

Shared Key → set box for automatically generation

Tunnel settings:

IPv4 Tunnel : 10.0.8.0/24 – сеть для виртуального соединения между OpenVPN

IPv4 Remote Network : 192.168.22.0/24

Настраиваем правила фильтрации межсетевого экрана:

Firewall → Rules

на вкладке WAN добавляем новое правило для входящих пакетов:

add rule

protocol → set UDP

Destination → set WAN address

Destination Port → set OpenVPN (1194)

На вкладке OpenVPN добавляем новое правило для входящих пакетов:

add rule

protocol → set any

Переходим назад в VPN->OpenVPN

нажимаем «e» для редактирования OpenVPN сервера и копируем общий ключ (Shared Key) в буфер обмена:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
6824dbad7bffe2563aa5ea2ba02b9fd2
82e53f384925d5bd3d0d8da53b339211
42611b47cd010d9b034af0b33128b593
5c03ab9b9116fcd223c99149db8adfe8
27d8a4a32e21608d47ca897785532f84
d62cec087e87bfd0be748add2c2e918e
9146fdab8f836039c3a253650b45cad1
cf8bcc2f31c00f4d6981c951443a3860
30a4c43a9b9916f73dbe658c2d4d7d37
c8415742b73dc716ffa5d908ef838d76
9093c31455eef85862a541bf2c583c7f
178d391551501b189d46f0e44f34a4eb
197ad863863b66dcaac42545063df653
aba1027a8225c330cd21352766221b35
49bad0bc7cfad90327d505c4d7ed8836
33a89fc0d3e90caadf69f3f777a92ae8
-----END OpenVPN Static key V1-----
```

Преходим к настройке клиента.

3. Развёртывание виртуального сегмента vnet2.

Аналогично развёртыванию виртуального сегмента vnet1, только необходимо указать соответствующие IP-адреса интерфейсов.

4. Настройка «pfsense_2» через web-интерфейс управления.

Настраиваем правила фильтрации межсетевого экрана аналогично «pfsense_1».

После этого, в меню VPN – OpenVPN «pfsense_2» переходим на вкладку Client и нажимаем кнопку “+” для создания клиента:

Server Mode → to Peer to Peer (Shared Key) (PSK)

Server host or address → 10.131.88.105 (адрес внешнего интерфейса pfsense с OpenVPN сервером)

Снять опцию «automatically generate a shared key» и вставить общий ключ, скопированный ранее с OpenVPN сервера.

Указать сеть для туннеля и удалённую сеть:

IPv4 Tunnel Network → 10.0.8.0/24

IPv4 Remote Network/s → 192.168.11.0/24

Проверить соединение с OpenVPN сервером, для этого смотрим информацию во вкладке «Status – OpenVPN». Если соединение не устанавливается, то переходим на сервере (pfsense_1) во вкладку

Status → System Logs → Firewall

видим, что от 10.131.88.106:44545 блокируются входящие соединения.

Нажимаем на красный крест в событии для просмотра дополнительной информации и получаем информацию вида

The rule that triggered this action is:

@62(1000001584) block drop in log quick on em0 inet from 192.168.0.0/16 to any label "Block private networks from WAN block 192.168/16"

Где поясняется, что блокировка идёт правилом «Block private networks».

Переходим в меню «Firewall: Rules» и видим, что удалить правило «Block private networks» не можем, но после нажатия «edit rules», переходим на вкладку, где его можно отключить.

Аналогично удаляем блокирующее правило на клиенте.

5. Проверка VPN-соединения.

Из виртуальной сети «vnet_1» проверяем прохождение пакетов в сеть «vnet_2». Для этого на виртуальной машине «ubuntu_min» сети «vnet_1» запускаем команду:

ping 192.168.22.5

для проверки связи с «ubuntu_min» сети «vnet_2».

Требования к содержанию и оформлению отчётов о выполнении лабораторных работ:

- отчёт должен содержать титульный лист с указанием названия работы, фамилий выполнивших студентов, номера группы (подгруппы, бригады), фамилии преподавателя; цель лабораторной работы; перечень используемых в лабораторной работе программных и аппаратных средств; ход работы со всеми выполнения задания в текстовом или графическом виде (снимки экранов монитора); результаты выполнения задания, выводы по лабораторной работе;
- отчёт должен быть оформлен в соответствии с требованиями к оформлению текстовых документов (ГОСТ 7.32-2001, ГОСТ 2.105-95).

Критерии оценивания лабораторных работ

Согласно учебной карте дисциплины на лабораторные работы по модулю 2 отводится 40 баллов. Каждая лабораторная работа оценивается в максимум 10 баллов. При оценивании лабораторной работы принимается во внимание активное участие в выполнении лабораторной работы; соблюдение требований к содержанию и оформлению отчёта о выполнении лабораторной работы, ответы на вопросы в процессе защиты отчёта о выполнении лабораторной работы.

Условием допуска к защите отчёта о выполнении лабораторной работы является факт выполнения задания лабораторной работы (в рамках аудиторных лабораторных занятий). Обучающийся, не выполнявший лабораторную работу, получает по ней 0 (ноль) баллов.

Оценивание каждой лабораторной работы проводится во время процедуры защиты отчёта о выполнении работы. Оценивание каждой лабораторной работы (в пределах 10 баллов) осуществляется по следующим элементам оценивания:

- до 4 баллов – оценивание выполнения работы и пояснений обучающимся хода выполнения работы;

- до 4 баллов – оценивание отчёта о выполнении лабораторной работы и пояснений обучающимся содержимого отчёта о выполнении лабораторной работы;
- до 2 баллов – оценивание ответа обучающегося на вопросы, связанные с принципами работы используемых в лабораторной работе средств.

Допускается каждый элемент оценивания лабораторной работы оценивать дробным числом баллов; после суммирования баллов по всем элементам оценивания набранная сумма баллов округляется до целого значения.

Критерии оценивания выполнения работы и пояснений обучающимся хода выполнения работы:

4 балла – обучающийся принимал активное участие в выполнении работы, может дать пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы);

2,1-3,9 балла – обучающийся принимал участие в выполнении работы, может дать частичные пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы); ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-2,0 балл – обучающийся принимал участие в выполнении работы, может дать частичные пояснения хода выполнения работы (как во время выполнения, так и на последующих занятиях при защите отчёта о выполнении работы); обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – обучающийся не может дать даже частичные пояснения хода выполнения работы, несмотря на участие в выполнении работы (присутствии на лабораторном занятии).

Критерии оценивания отчёта о выполнении лабораторной работы и пояснений обучающимся содержимого отчёта о выполнении лабораторной работы:

4 балла – отчёт о выполнении работы содержит все требуемые разделы; отчёт соответствует заданному варианту; отчёт оформлен в соответствии с требованиями к оформлению текстовых документов (ГОСТ 7.32-2001, ГОСТ 2.105-95); отчёт содержит все требуемые этапы выполнения задания; отчёт содержит все требуемые результаты выполнения задания; обучающийся может дать полные пояснения любого места отчёта о выполнении лабораторной работы, в том числе полученных результатов;

2,1-3,9 балла – отчёт о выполнении работы содержит все требуемые разделы; отчёт соответствует варианту выполнения экспериментальных исследований; отчёт оформлен в соответствии с требованиями к оформлению текстовых документов; отчёт содержит все требуемые этапы выполнения задания; отчёт содержит большинство требуемых результатов выполнения задания; обучающийся может дать частичные пояснения отдельных фрагментов отчёта о выполнении лабораторной работы, в том числе полученных результатов; ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-2,0 балл – отчёт о выполнении работы содержит основные требуемые разделы (в том числе ход работы и обработку результатов); отчёт соответствует варианту задания; отчёт в целом оформлен в соответствии с требованиями к оформлению текстовых документов без грубых нарушений требований; отчёт содержит большинство требуемых этапов выполнения задания; отчёт содержит большинство требуемых результатов выполнения задания; обучающийся может дать частичные пояснения отдельных фрагментов отчёта о выполнении

лабораторной работы, в том числе полученных результатов; обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – либо отчёт о выполнении работы не содержит все требуемые разделы; либо отчёт не соответствует варианту задания; либо отчёт оформлен с грубыми нарушениями требований к оформлению текстовых документов; либо отчёт не содержит большинства этапов выполнения задания; либо отчёт не содержит большинства требуемых результатов выполнения задания; либо обучающийся не может дать даже частичные пояснения отдельных фрагментов отчёта о выполнении лабораторной работы, в том числе полученных результатов.

Критерии оценивания ответа обучающегося на вопросы, связанные с принципами работы используемых в лабораторной работе программных средств и методов:

2 балл – обучающийся может дать полные ответы на вопросы, связанные с принципами работы используемых в лабораторной работе программных средств и методов;

1-1,9 балла – обучающийся может дать частично верные ответы на вопросы, связанные с принципами работы используемых в лабораторной работе программных средств и методов; ошибки и неточности в ответах самостоятельно исправляются обучающимся в ходе ответов на дополнительные вопросы;

0,1-0,9 балла – обучающийся может дать частично верные ответы на вопросы, связанные с принципами работы используемых в лабораторной работе программных средств и методов; обучающийся испытывает затруднения в самостоятельном исправлении ошибок и неточности в ответах;

0 баллов – обучающийся не может дать даже частично верные ответы ни на вопросы, связанные с принципами работы используемых в лабораторной работе программных средств и методов.