УНИВЕРСИТЕТ ИННОПОЛИС

INNOPOLIS OPEN

Международная олимпиада «Innopolis Open» по профилю «Информационная безопасность»

Материалы заданий заключительного этапа олимпиады

2024-2025

Первый тур заключительного этапа

Первый тур заключительного этапа прошел на специальной платформе в формате CTF task-based с добавлением творческих задач.

Особенности тура:

- Длительность тура: 4 астрономических часа.
- Интернет отсутствует.
- Для каждого участника был подготовлен ноутбук с предустановленной ОС Kali Linux
- Каждая задача по категориям оценивалась в 1 балл.
- Задания из категорий «Творческое» и «Бумажное» оценивались от 0 до 2 баллов с шагом 0,25 балла.
- Задания из категории «Бумажное» решались на выданных бланках.
- Платформа хостилась на локальной сети Университета, что ограничивало поиск подсказок в сети интернет.

Все задания можно отнести к следующим категориям:

- **ADMIN** (задания на администрирование операционных систем). Обычно задания, связанные с работой сисадмина: восстановление данных, виртуальные машины и так далее.

- **CRYPTO** (Криптография – задания на криптографические алгоритмы, как на старинные, так и на современные).

- **FORENSIC** (Компьютерная криминалистика – расследование инцидентов, исследование различных дампов (сетевых, памяти и прочее), восстановление архивов.

- **REVERSE (**Обратная разработка – исследование бинарных файлов (программ) без исходных кодов и изучение работы различных редких архитектур)

- **WEB** (Поиск и эксплуатация веб-уязвимостей)

Задания Первого тура заключительного этапа

1. Творческое

1.1. Кибервойны будущего: какими будут атаки на критическую инфраструктуру?

Балл: 2

Условие:

Если хакеры отключат энергосети или системы водоснабжения целого города, как общество должно реагировать? Какие превентивные меры (например, воздушные зазоры, резервные аналоговые системы) актуальны в эпоху цифровизации? Должны ли кибератаки приравниваться к военным действиям и регулироваться международными конвенциями?

Требования к ответу:

1. Ваш ответ должен быть подробным и аргументированным.
2. По возможности, подкрепите свои утверждения примерами.
3. Ожидается критический анализ представленных данных и мнений.

Формат сдачи:

Ответ предоставляется в форме эссе. Ожидается чёткое структурирование текста, наличие введения, основной части и заключения. Ответ сохранять на рабочем столе с названием "творческая задача 1"

1.2. Анонимность в сети: утопия или необходимость?

Балл: 2

Условие:

Tor, криптовалюты и E2E-шифрование позволяют сохранять анонимность, но также используются для незаконной деятельности. Должно ли общество пожертвовать анонимностью ради безопасности? Может ли существовать система, где права на приватность не противоречат борьбе с преступностью?

Требования к ответу:

1. Ваш ответ должен быть подробным и аргументированным.
2. По возможности, подкрепите свои утверждения примерами.
3. Ожидается критический анализ представленных данных и мнений.

Формат сдачи:

Ответ предоставляется в форме эссе. Ожидается чёткое структурирование текста, наличие введения, основной части и заключения. Ответ сохранять на рабочем столе с названием "твроческая задача Incident Manager"

2. Paper

2.1. Crypto

Балл: 1.5

Условие:

Злоумышленник зашифровал файл с помощью XOR. Известно:

- Ключ: 3 байта.
- Первые 3 байта исходного файла: PDF (в hex: 50 44 46).
- Первые 3 байта зашифрованного файла: 12 2B 24.

Вопросы:

1. Найдите ключ (в hex).
2. Расшифруйте байт 0x7F с помощью этого ключа.
3. Запишите все шаги вычисления XOR.

2.2. Ошибка в алгоритме

Балл: 1.5

Условие:

```
if password[0] == 'A' and password[3] == 'X' and len(password) == 5:
    print("Доступ разрешен")
else:
    print("Ошибка")
```

Вопросы:

1. Сколько всего паролей удовлетворяют условию (если пароль —заглавные буквы A-Z)?
2. Как можно эксплуатировать эту проверку для подбора пароля?
3. Предложите исправление кода для повышения безопасности.

3. Admin

3.1. Больше не друзья

Балл: 1

Условие:

Раньше мы с Андреем были друзья. Сидели за одной партой, играли в одной футбольной команде. Но однажды он меня предал, возможно, начал завидовать, что я победил на олимпиаде по математике. Мы делали финальный проект по информатике, показывали навыки Linux и написание своего веб-сервера. Я, признаюсь, скачал работу с Интернета, немного её доделал и опубликовал на сервере. Андрей также имел доступ к этому серверу, он удалил мой ssh-ключ и что-то модифицировал в коде моего сервера, теперь не могу скачать документы и просмотреть их. Поможешь разобраться? В одном из документов есть очень важная информация, моя грамота победителя по математике. Достань хотя бы её

4. Forensic

4.1. Преступление и наказание

Балл: 1

Условие:

В детстве на уроках литературы мы переписывались зашифрованными цифрами. Уже не помню, что они значили, но для шифровки мы использовали учебник. Я недавно пересматривал книги на своей полке, наткнулся на бумажку с таким шифротекстом. Вот бы расшифровать, вспомнить, о чем же мы там болтали…

* 14:1:1
* 65:15:2
* 7:14:5
* 88:6:14
* 52:3:1
* 24:9:10

* 36:4:2
* 48:9:5
* 6:7:6
* 58:10:16

Ответом будет русское слово, существующее

Пример: аллигатор

Учитывать только текст книги, не учитываются заголовки, нижние и верхние колонтитулы

5. Web

Школьный альбом

Балл: 1

Условие:

Школьный альбом - замечательная память о своей молодости, о своих школьных товарищах. Наш альбом - пример всем. Ежегодно мы обновляем данные, пишем чем занимаются наши школьные друзья, вместе растем и достигаем успехов. К сожалению, год назад я потерял доступ к серверу, альбом перестал обновляться. Может ли быть такое, что сервер взломал хакер и закрыл доступ? Можешь ли мне скачать этот сайт, я перенесу его на другой хостинг? А хотя, постойте… У меня было 20 одноклассников, а на сайте всего 20 фото, не хватает одного. А что если хакер удалил какие-то данные? Сможешь восстановить? Я даже и не помню, кого удалили, совсем позабыл кого как зовут, редко видимся. Сообщи мне имя и фамилию человека, которого удалили, и скачай мне сайт для переноса. Спасибо!

6. Crypto

Солнечный язык

Балл: 1

Условие:

В детстве мы придумывали разные способы общения, чтобы дети с других районов не понимали, о чем мы говорим. Это, конечно дурачество, но такая коммуникация помогала нам "выделяться". Так появился "солнечный язык", кто-то называл это "нашим" языком, кто-то - "кирпичным". Суть там была одна:

после каждой гласной буквы ставишь букву (с) а после (с) ставишь букву ту, которая стояла перед этой буквой. Таким образом, фраза "Всем привет" превращалась в "Всесем присивесет".

Ваша задача - написать такой сервис, который позволяет

1. регистрироваться (/register) с почтой и безопасным паролем (безопасный - минимум 8 символов, минимум одна заглавная буква, минимум одна прописная, минимум один символ и минимум одна цифра).
2. входить (/login) под логином и паролем из пункта 1, и получать сессию (это cookie с названием session и любым уникальным значением). Время жизни - не более минуты
3. проверять информацию о себе (/info) по сессии - ip адрес и идентификатор сессии (неважно какой это идентификатор, строка, uuid, число)
4. делать перевод (/translate) на солнечный язык. На входе - строка (слово, предложение на русском языке). На выходе - строка на солнечном.

После написания такого сервиса, необходимо в специальную форму на сервере жюри отправить запрос на проверку. Мы сами определим ваш ip-адрес, на котором запущено приложение, порт строго должен быть 1337

Методика проверки жюри:

1. /login с несуществующими данными, должно дать ошибку
2. /register с неправильной почтой и/или паролем, должно дать ошибку
3. /register с правильной почтой и паролем, должно нас зарегистрировать в системе

4. /login с данными из пункта 3, в ответ получаем cookie с переменной session и вашим произвольным значением сессии (генерация на вас)
5. /info - мы получим информацию о своём ip-адресе и идентификаторе сессии (может совпадать со значением из пункта 4, может быть какое-то другое значение, например id в вашей системе)
6. мы сделаем 3 запроса на /translate. На входе подадим строку/предложение, на выходе вы отправите сообщение на солнечном языке. Мы проверим корректность перевода, если будет ошибка - проверка досрочно завершится
7. если все проверки пройдут, то на странице сервера жюри у вас появится флаг

В качестве примера прикладываем дамп сетевого трафика. В данном трафике показано взаимодействие между сервером, который реализует указанный выше функционал, с системой проверки жюри.

7. Reverse

7.1. LZ77

Балл: 1

Условие:

Напишите алгоритм декомпрессии (разархивирования) на основании документа и исходного кода сжатия. Получите оригинал файла и прочтите, что там было написано?

Пример компрессии

```
def lz77_compress(data):
    compressed = bytearray()        # Результат сжатия
    current_pos = 0                 # Текущая позиция в данных
    max_offset = 4095               # Максимальное смещение (12 бит)
    max_length = 16                 # Максимальная длина совпадения (4 бита → 1-16)

    # Пока не обработаны все данные
    while current_pos < len(data):
        bits = []                   # Бит управления (1 или 0 для 16 операций)
        commands = []               # Команды (ссылки или литералы)

        # Собираем 16 команд для текущего блока
        for _ in range(16):
            if current_pos >= len(data):
                # Если данные закончились, добавляем фиктивные литералы
                bits.append(0)
                commands.append(0x00)
            else:
                # Ищем лучшее совпадение в предыдущих данных
                best_offset, best_length = find_best_match(data, current_pos, max_offset,
max_length)
```

```
        if best_length >= 3:
            # Кодируем ссылку
            bits.append(1)
            commands.append((best_offset, best_length))
            current_pos += best_length
        else:
            # Кодируем литерал
            bits.append(0)
            commands.append(data[current_pos])
            current_pos += 1

# Формируем управляющее слово (16 бит)
control_word = 0
for i in range(16):
    control_word |= (bits[i] << i)  # Младший бит соответствует первой операции

# Добавляем управляющее слово в little-endian
compressed.extend(control_word.to_bytes(2, 'little'))

# Добавляем команды
for cmd in commands:
    if isinstance(cmd, tuple):
        # Кодируем ссылку: 2 байта
        offset, length = cmd
        # Байт 1: старшие 4 бита смещения + длина-1
        byte1 = ((offset >> 8) << 4) | ((length - 1) & 0x0F)
        # Байт 2: младшие 8 бит смещения
        byte2 = offset & 0xFF
```

```
            compressed.append(byte1)
            compressed.append(byte2)
        else:
            # Литерал: 1 байт
            compressed.append(cmd)

    return compressed

def find_best_match(data, current_pos, max_offset, max_length):
    if current_pos >= len(data):
        return (0, 0)

    best_length = 0
    best_offset = 0
    start = max(0, current_pos - max_offset)

    # Перебираем возможные смещения (от 1 до max_offset)
    for offset in range(1, current_pos - start + 1):
        match_len = 0

        # Сравниваем символы пока есть совпадения
        while (current_pos + match_len < len(data) and
               match_len < max_length and
               data[current_pos - offset + match_len] == data[current_pos + match_len]):
            match_len += 1

        # Обновляем лучшее совпадение
        if match_len > best_length:
            best_length = match_len
```

```
            best_offset = offset

    # Минимальная длина совпадения - 3 символа
    return (best_offset, best_length) if best_length >= 3 else (0, 0)

# Пример данных с повторениями
with open('original.pdf', 'rb') as f:
    data = f.read()
# Сжимаем
compressed_data = lz77_compress(data)

# Сохраняем в файл
with open("compressed.bin", "wb") as f:
    f.write(compressed_data)
```

Алгоритм декомпрессии

Декодирование (распаковка)

Шаги алгоритма:

1. Инициализация:
    - Создать пустой буфер для распакованных данных.
    - Установить текущую позицию (offset) в начало сжатых данных.
2. Чтение управляющего слова:
    - Прочитать 16 бит (2 байта) — это управляющее слово.
    - Каждый бит управляющего слова определяет тип команды (1 — ссылка, 0 — литерал).
3. Обработка команд:
    - Для каждого бита управляющего слова:
        - Если бит = 1:
        - Прочитать 2 байта ссылки.
        - Декодировать смещение и длину.

▪ Скопировать данные из буфера распакованных данных по указанному смещению.

▪ Если бит = 0:

▪ Прочитать 1 байт литерала.

▪ Добавить его в буфер распакованных данных.

4. Завершение:

o Если данные закончились, завершить процесс.

Пример декодирования:

Входные данные: Управляющее слово 0x0001, ссылка (7, 4), литералы a, b, c.

5. Прочитать управляющее слово 0x0001 (первая команда — ссылка).

6. Декодировать ссылку (7, 4) и скопировать abra.

7. Прочитать литералы a, b, c.

8. Результат: abracadabra.

8. Web

8.1. Топ рега

Балл: 1

Условие:

В рамках обеспечения суверенитета Российской Федерации в телекоммуникационном пространстве, всем гражданам необходимо пройти обязательную регистрацию в новой социальной сети "Береста". После проверки вашей почты на госуслугах, вам станут доступны все функции, а пока оставайтесь на странице профиля