

Решение Reverse-9 Забытый рецепт крафта

Описание

В древних руинах ты обнаружил таинственный свиток с рецептом легендарного артефакта. Но кто-то разбросал ингредиенты рецепта по разным сундукам в соседних комнатах. Найди все компоненты и узнай секрет древнего крафта!

Решение

Шаг 1: Анализ структуры задания

Скачиваем исходные файлы и смотрим в основной файл с Python кодом - `task.py`. Видим множество импортов из разных "сундуков" (пакетов):

```
from wooden_chest.old_map import TREASURE_X, MAP_DATA

from diamond_chest.ancient_scroll import RECIPE_FRAGMENT_ALPHA

from emerald_chest.fake_ingredient import USEFULL_DATA, DECOY_VALUE

from iron_chest.crafting_table import RECIPE_FRAGMENT_BETA

from gold_chest.enchanted_book import ENCHANTMENT_LEVEL, MAGIC_RECIPE

from netherite_chest.nether_portal import RECIPE_FRAGMENT_GAMMA
```

Всего 6 различных сундуков! Нужно понять, какие из них содержат настоящие фрагменты рецепта (флага).

Шаг 2: Исследование содержимого сундуков

Заходим в каждую папку и смотрим содержимое файлов:

wooden_chest/old_map.py - деревянный сундук:

```
TREASURE_X = 1234
```

```
TREASURE_Z = 5678
```

```
MAP_DATA = "Tm90aGluZyB1c2VmdWwgaGVyZQ=="
```

Похоже на координаты карты - не похоже на фрагмент флага.

diamond_chest/ancient_scroll.py - алмазный сундук:

```
RECIPE_FRAGMENT_ALPHA = "L4YGMX3UNAZQ===="
```

Название переменной `RECIPE_FRAGMENT_*` похоже на фрагмент флага.

emerald_chest/fake_ingredient.py - изумрудный сундук:

```
USEFULL_DATA = "RmFrZSBjbmdyZWRpZW50"
```

```
DECOY_VALUE = 42
```

Название файла `fake_ingredient` намекает, что это ложные данные.

iron_chest/crafting_table.py - железный сундук:

```
RECIPE_FRAGMENT_BETA = "5F726563653170747D"
```

Также как с ALPHA похоже на фрагмент флага.

gold_chest/enchanted_book.py - золотой сундук:

```
ENCHANTMENT_LEVEL = "III"
```

```
MAGIC_RECIPE = "VGhpcyBpcyBub3QgdGhlIHJlY2lwZSB5b3UgYXJlIGxvb2tpbmcgZm9y"
```

Зачарованная книга - скорее всего отвлекающий маневр.

netherite_chest/nether_portal.py - незеритовый сундук:

```
RECIPE_FRAGMENT_GAMMA = "dnNvc2h7czNjcjN0"
```

Третий фрагмент рецепта (флага)

Шаг 3: Определение типа кодирования

Нашли три фрагмента:

- `RECIPE_FRAGMENT_ALPHA = "L4YGMX3UNAZQ===="`
- `RECIPE_FRAGMENT_BETA = "5F726563653170747D"`
- `RECIPE_FRAGMENT_GAMMA = "dnNvc2h7czNjcjN0"`

По характерным признакам определяем кодировку:

- **ALPHA**: заканчивается на `====` → Base32
- **BETA**: только hex символы (0-9, a-f) → Hex (Base16)
- **GAMMA**: стандартный Base64 формат

Шаг 4: Декодирование и сборка флага

В Python есть встроенная библиотека `base64` с методами для всех нужных кодировок. Пишем скрипт-решение:

```
import base64

# Фрагменты рецепта из сундуков
RECIPE_FRAGMENT_GAMMA = "dnNvc2h7czNjcjN0" # из netherite_chest

RECIPE_FRAGMENT_ALPHA = "L4YGMX3UNAZQ====" # из diamond_chest

RECIPE_FRAGMENT_BETA = "5F726563653170747D" # из iron_chest

# Декодируем GAMMA (Base64)
fragment_1 = base64.b64decode(RECIPE_FRAGMENT_GAMMA).decode()
print(f"Фрагмент из Netherite сундука (Base64): {fragment_1}")

# Декодируем ALPHA (Base32)
fragment_2 = base64.b32decode(RECIPE_FRAGMENT_ALPHA).decode()
print(f"Фрагмент из Diamond сундука (Base32): {fragment_2}")

# Декодируем BETA (Hex)
```

```
fragment_3 = bytes.fromhex(RECIPE_FRAGMENT_BETA).decode()
print(f"Фрагмент из Iron сундука (Hex): {fragment_3}")

# Собираем легендарный артефакт (флаг)
legendary_artifact = fragment_1 + fragment_2 + fragment_3
print(f"\nЛегендарный артефакт создан: {legendary_artifact}")
```

Шаг 5: Результат

Запускаем скрипт и получаем:

Флаг: `vsosh{s3cr3t_of_th3_rece1pt}`