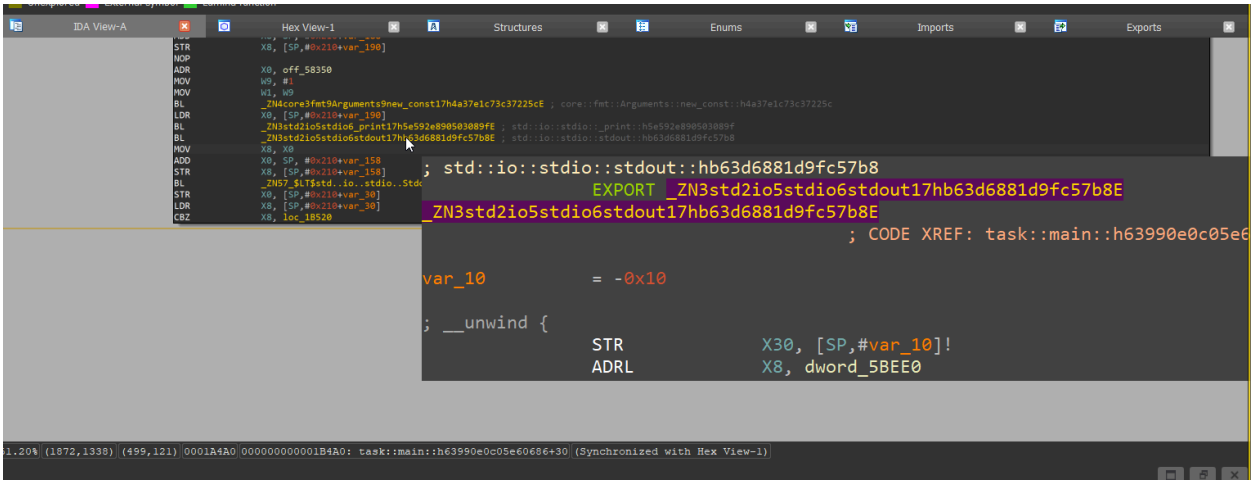
Название:	Ржавый ARM
Категория:	Реверс-инжиниринг
Уровень:	Средняя
Очки:	350
Описание:	Цель проста - получить флаг :)
Теги:	Rust, ARM
Автор:	ROP

Прохождение:

Откроем файл в IDA Pro.



Видим, что это Rust под ARM64.

Попробуем его восстановить в отладке или статике и получаем примерно такой код:

```
use std::io::{self, Write};
use std::process;

fn main() {
    print!("Enter the flag: ");
    io::stdout().flush().unwrap();
    let mut password = String::new();

    if io::stdin().read_line(&mut password).is_err() {
```

```
        eprintln!("Failed to read the flag :(");
        process::exit(1); // Завершаем программу с кодом ошибки
    }

    password = password.trim().to_string();

    let encrypted_password: Vec<u8> = password.bytes().map(|b| {
        let add_result = b.wrapping_add(0x30);
        let xor_result = add_result ^ 0x45;
        let sub_result = xor_result.wrapping_sub(0x99);
        let qwe = sub_result.wrapping_add(0x35);
        let xor_result2 = qwe ^ 0x45;
        xor_result2
    }).collect();

    let reference_password: Vec<u8> = vec![
        0x97, 0x93, 0x88, 0x89, 0x96, 0x2d, 0xcf, 0x26, 0x86, 0xc7, 0x18, 0x23, 0x2a, 0x84, 0x26, 0x23,
        0x95, 0x26, 0x91, 0x23, 0x3b, 0x3c, 0x94, 0x38, 0x83, 0xc1,
    ];

    if encrypted_password == reference_password {
        println!("It's a flag!!!");
    } else {
        println!(":(");
    }
}
```

Для каждого байта в `reference_password` применяем обратные операции:

1. XOR 0x45
2. SUB 0x35
3. ADD 0x99
4. XOR 0x45
5. SUB 0x30

https://cyberchef.org/#recipe=From_Hex('Auto')XOR({'option':'Hex','string':'45'},'Standard',false)