SOC
FORUM
2023

Фантастические инструменты анализа ВПО и как их использовать

SOC
FORUM
2023

Нестор Владимир

Руководитель группы анализа ВПО
центра исследования киберугроз
Solar 4RAYS

ГК «СОЛАР»

Фантастические инструменты анализа ВПО

Rizin

Emulation

SMT Solver

Symbolic execution

ЗАЧЕМ ИХ ИСПОЛЬЗОВАТЬ

1. Быстрая реакция во время инцидента
2. Подробный анализ ВПО
3. Доступный для освоения любимым специалистом

Rizin

Инструмент Rizin

SOC FORUM 2023

ФУНКЦИИ

- Дизассемблер
- Отладчик

ОСОБЕННОСТЬ

- Форк radare2
- Фрейморк для автоматизации

ВПО Meterpreter

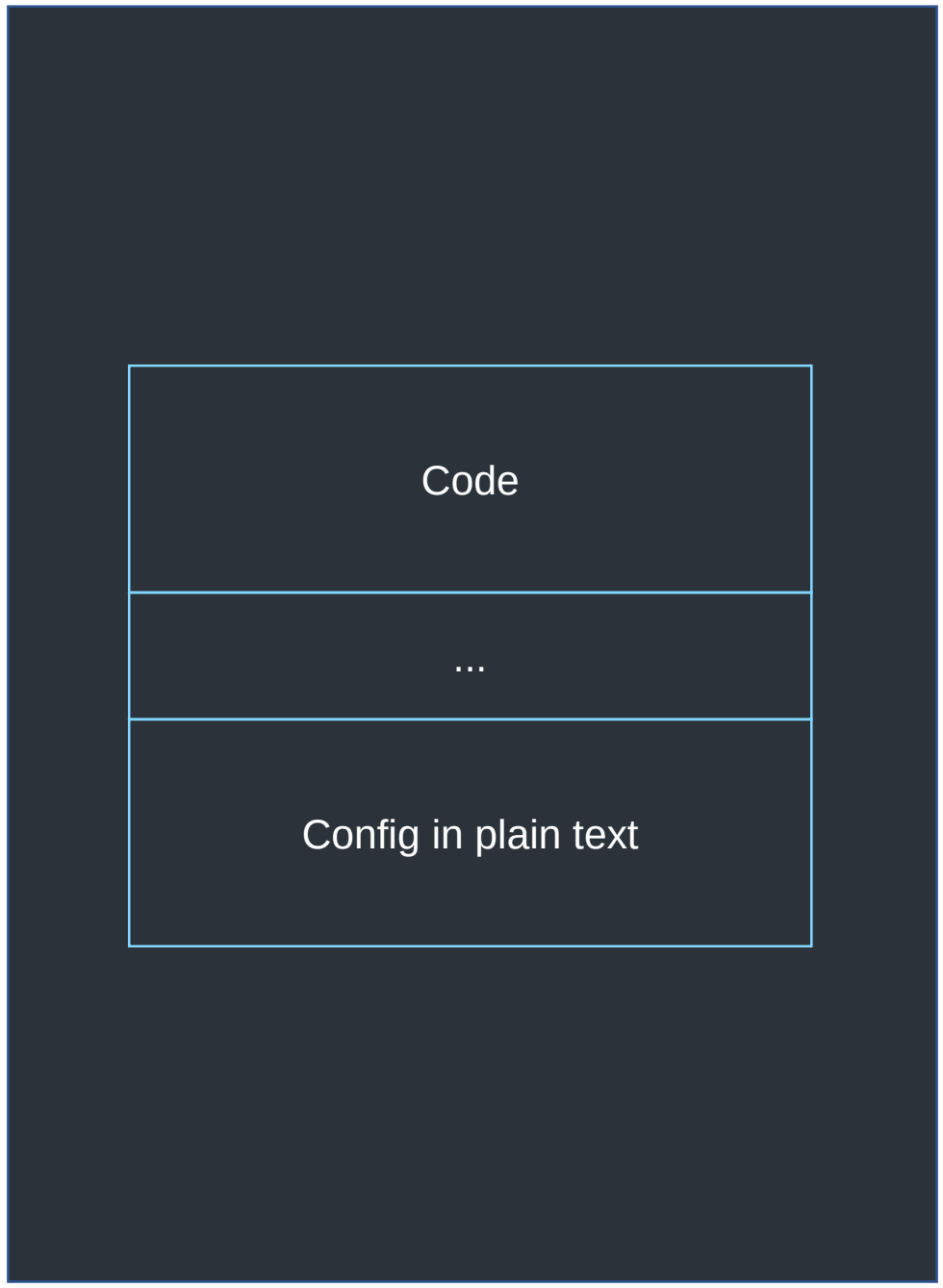
SOC FORUM 2023

1 Улучшенная версия шелла для метасплойта

2 Используется в связке с другими инструментами

3 Часто упакован и защищен

ВПО Meterpreter



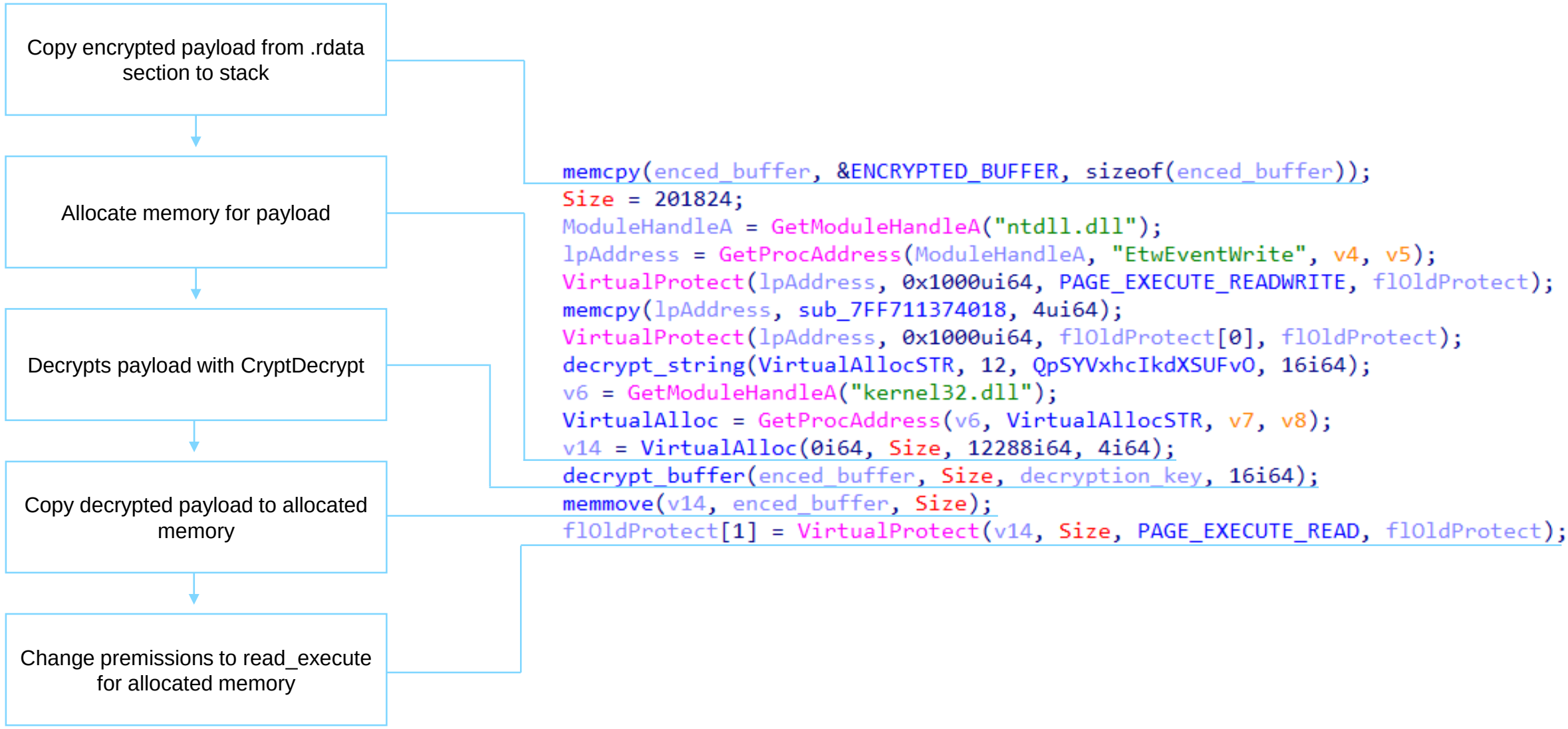
1 Полезная нагрузка зашифрована и расшифровывается лоадером в памяти

2 Полезная нагрузка зашифрована и расшифровывается лоадером в памяти

Проблема: а как достать в автоматизированном виде C2?

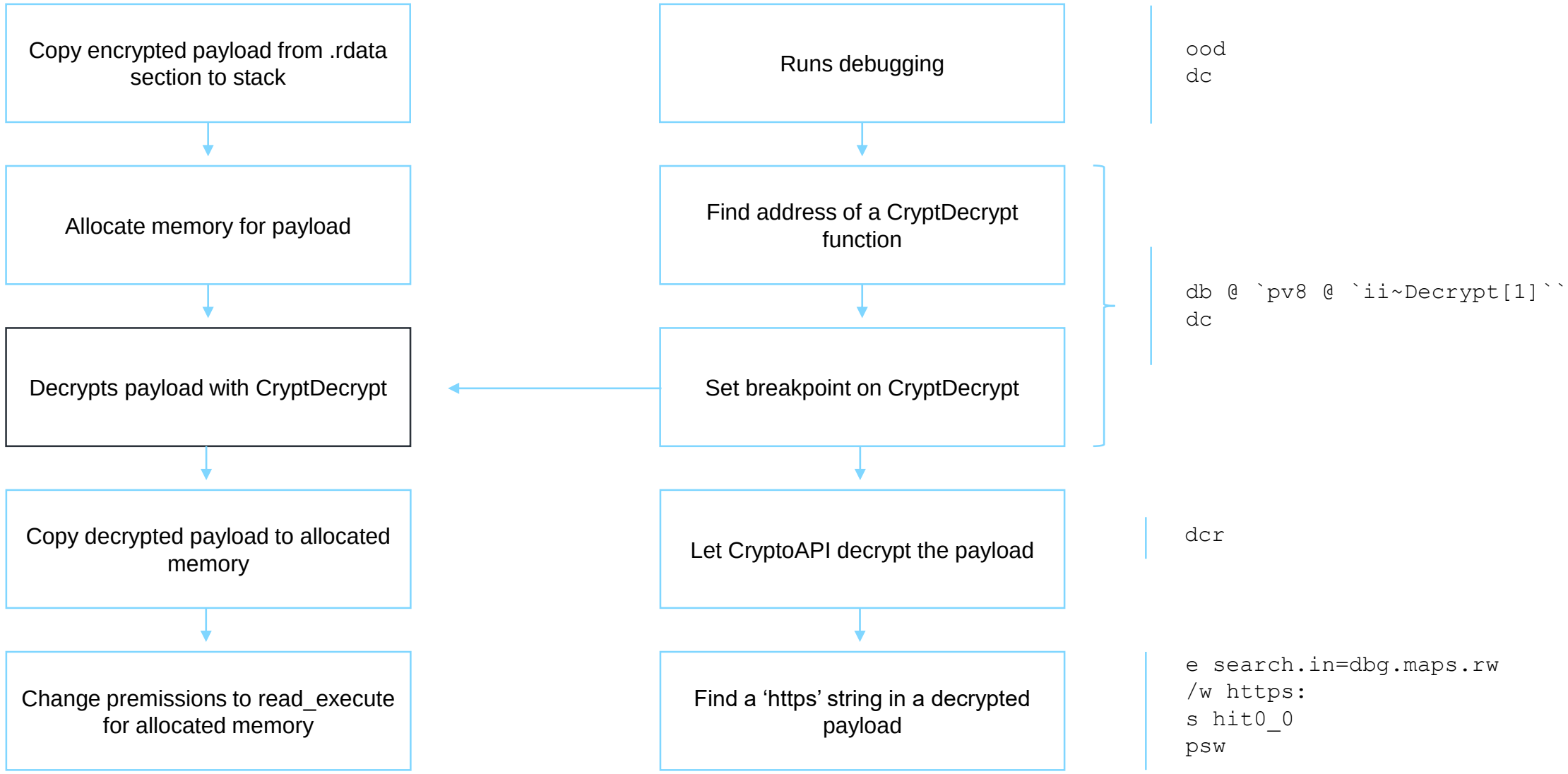
Как работает загрузчик Meterpreter

SOC FORUM 2023



Автоматизация сбора информации из Meterpreter

SOC FORUM 2023

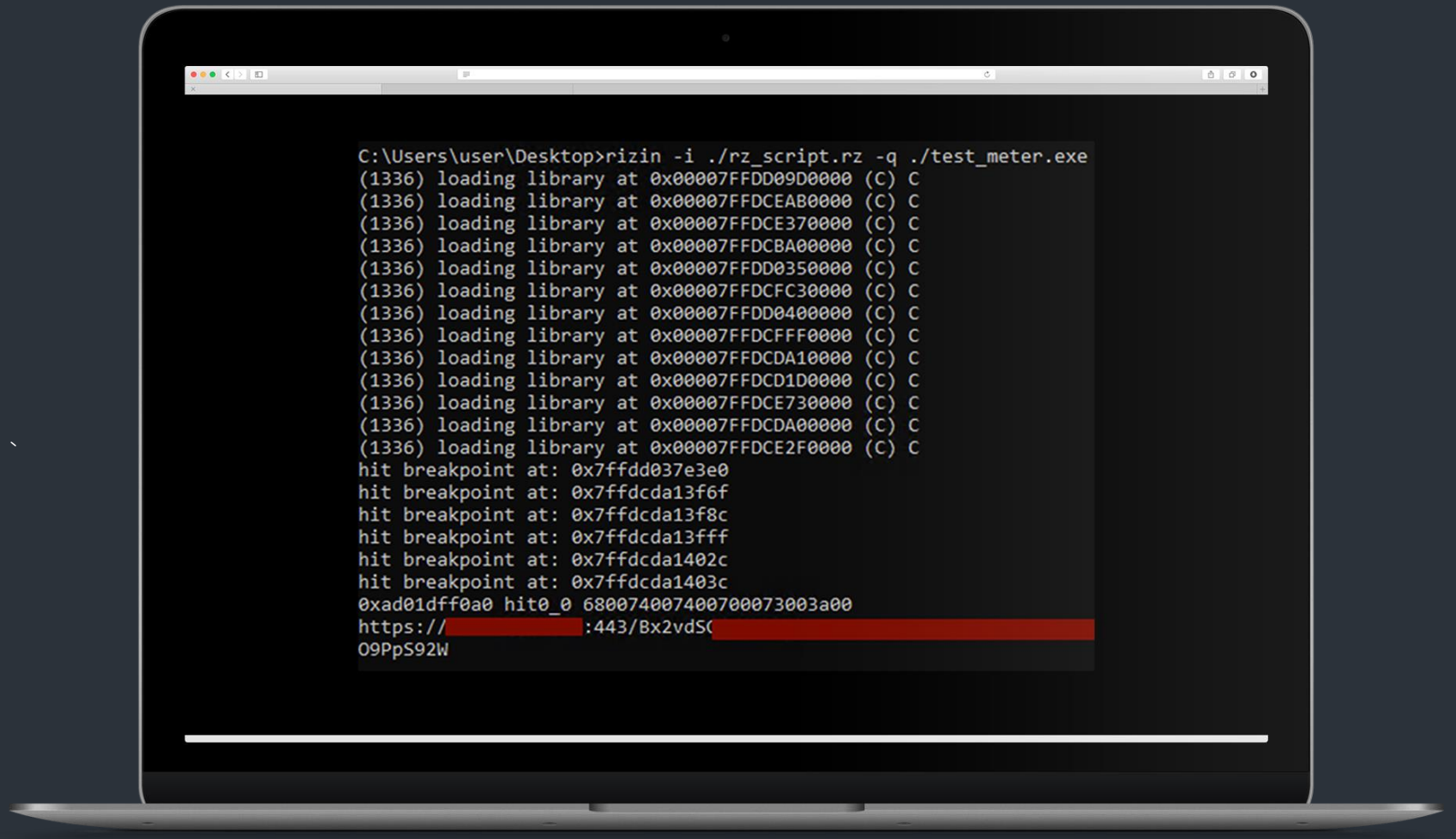


Инструмент Rizin в действии

SOC FORUM 2023

```
rizin -q -i rz_script.rz
test_meter.exe

rz_script.rz:
  ood
  dc
  db @ `pv8 @ `ii~Decrypt[1]``
  dc
  dcr
  e search.in=dbg.maps.rw
  /w https:
  s hit0_0
  psw
```



Unicorn

Инструмент Unicorn

SOC FORUM 2023

ФУНКЦИИ

- Эмулятор кода

ОСОБЕННОСТЬ

- Позволяет полностью контролировать выполнение кода программным способом

ВПО ShadowPad Light

SOC FORUM 2023

1. Известен также под названием Deed RAT
2. Используется группировкой Space Pirates
3. До сих пор активно
4. Конфигурация также шифрована и хранится в реестре
5. Полезная нагрузка зашифрована и расшифровывается лоадером в памяти

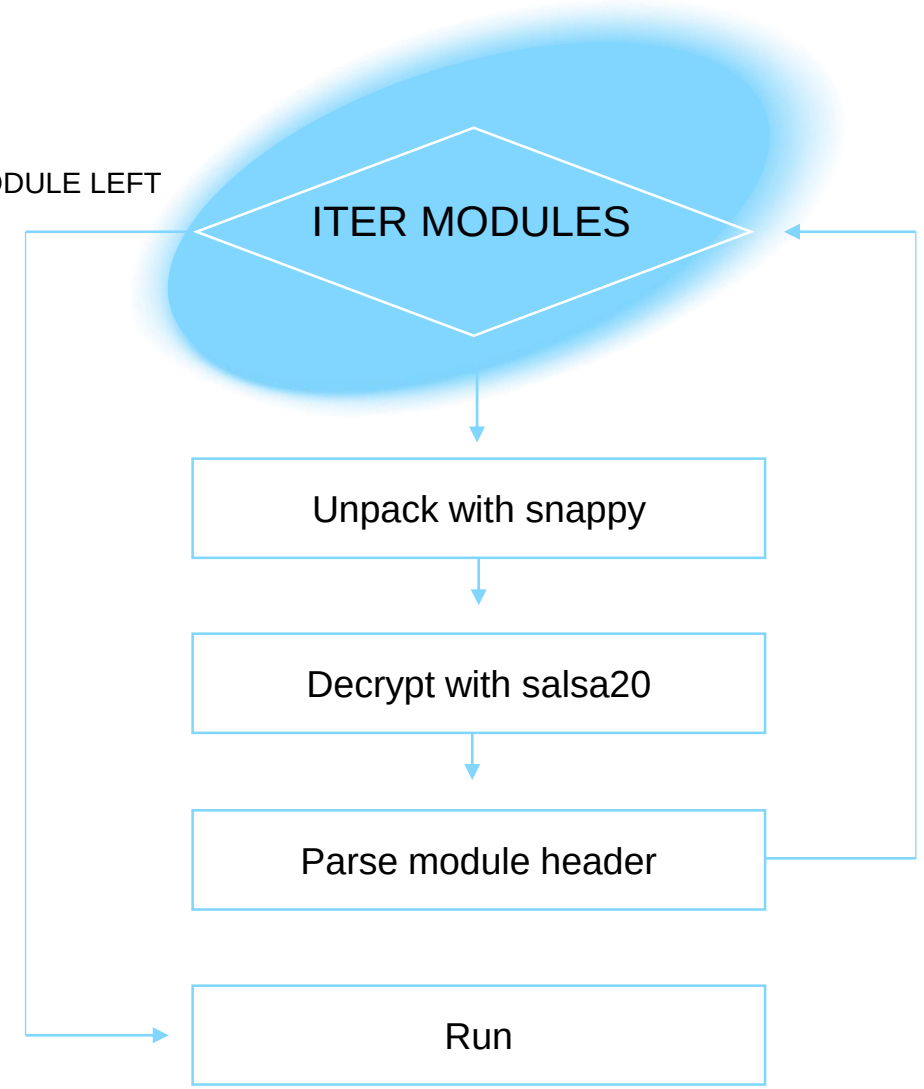
Как работает модуль загрузки ShadowPad Light

SOC FORUM 2023

ИСТОЧНИК МОДУЛЕЙ:

- Встроенные в главную полезную нагрузку
- Хранятся в реестре

Конфигурация хранится в реестре, зашифрованная тем же алгоритмом



Варианты расшифровки модулей ShadowPad Light

SOC FORUM 2023

1

Разобрать алгоритм шифрования и распаковки

2

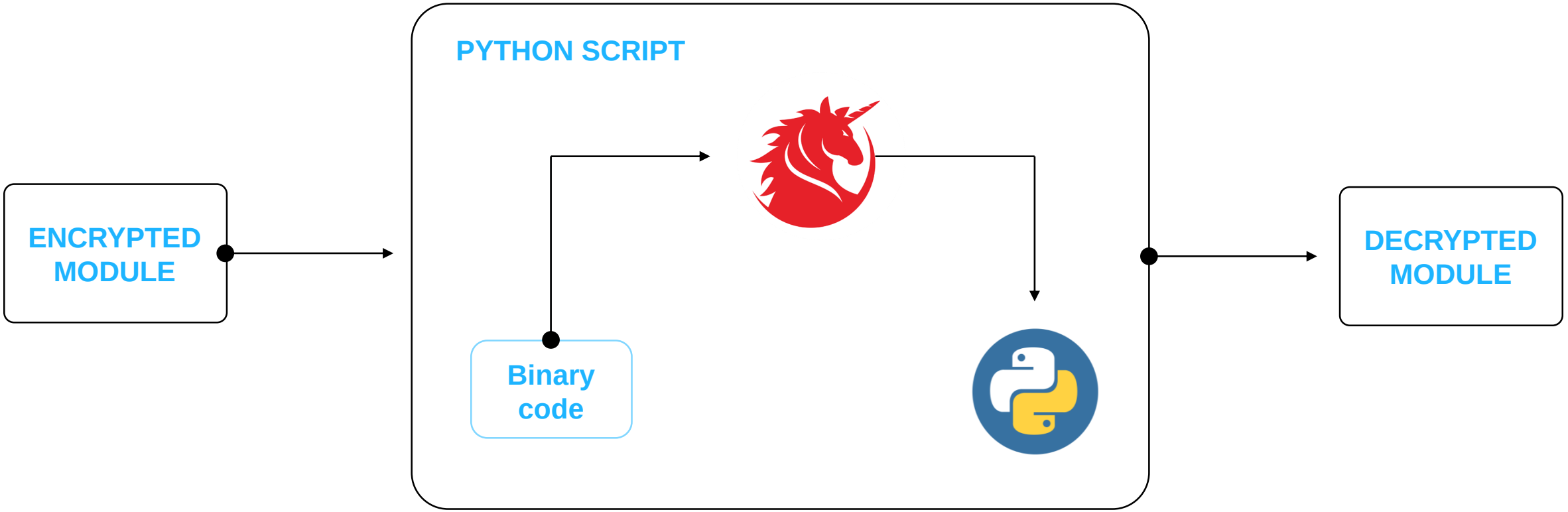
Сконфигурировать окружение и расшифровать в динамике

3

Применить эмуляцию кода

ShadowPad Light + Unicorn = <3

SOC FORUM 2023



ShadowPad Light + Unicorn = <3

SOC FORUM 2023

crypt_shellcode.py

```
_f_cryption = [ 0xDE, 0xAD,...
              ]
_main_crypt = [ 0xBE, 0xEF,...
              ]
_the_key    = [ 0xCA, 0xFE,...
              ]
_f_cryption_offset = 0xdead
_main_crypt_offset = 0xbeef
_the_key_offset    = 0xcafe
```

decrypt.py

```
import crypt_shellcode

init_unicorn()    # 4 lines
setup_memory()    # 22 lines
setup_registers() # 8 lines

run_function()    # 1 line
read_memory()     # 1 line
```

Еще один инструмент: Flare-emu

SOC FORUM 2023

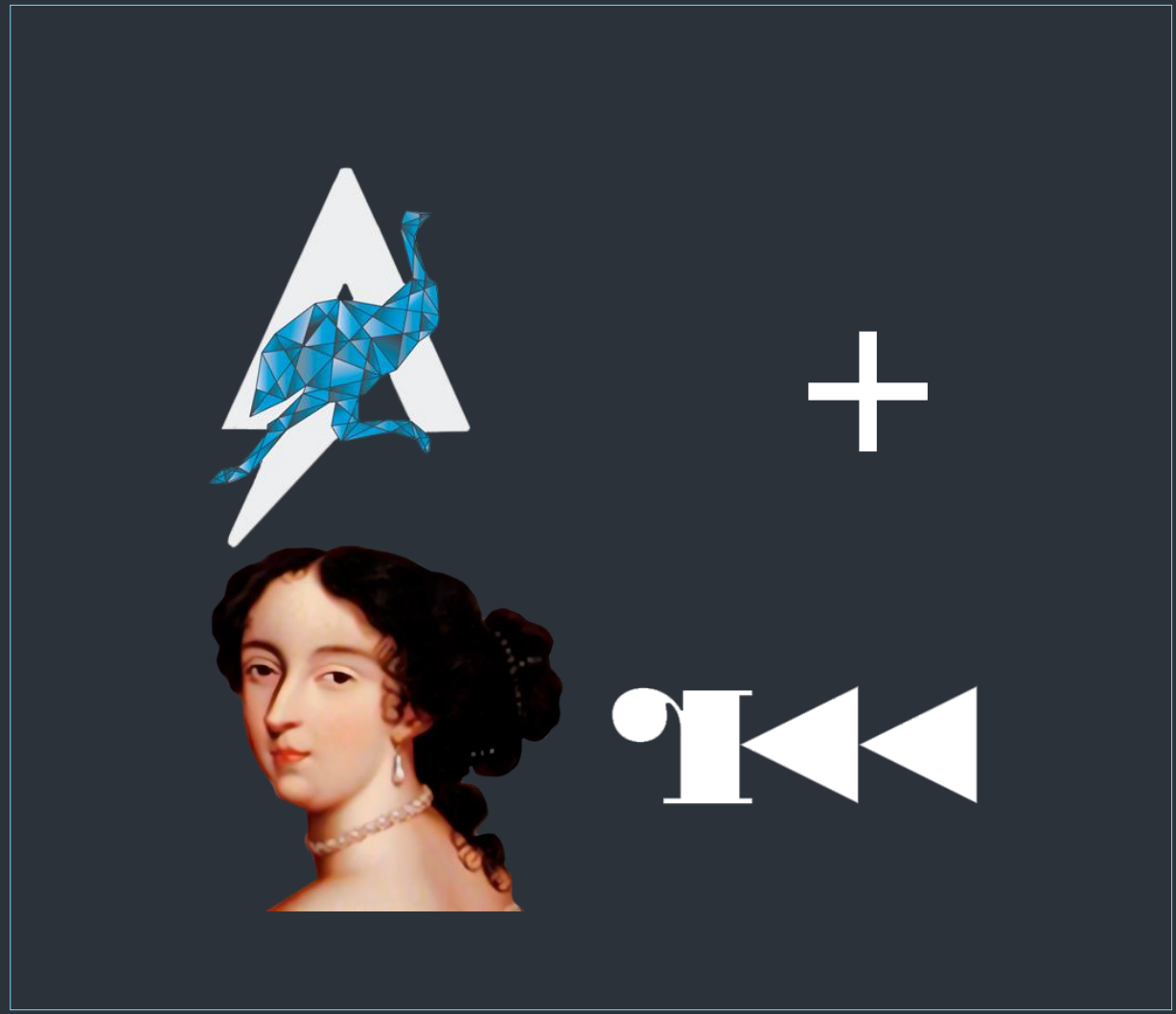
ФУНКЦИИ

- Эмулятор кода

ОСОБЕННОСТЬ

Обертка над unicorn

- Позволяет легко эмулировать отдельные функции внутри IDA Pro
- Связь с радаром позволяет создавать почти независимые скрипты для обработки чего угодно
- Содержит функции, эмулирующие работу VirtualAlloc и подобных



Как Flare-emu расшифровывают строки лоадера Meterpreter

SOC FORUM 2023

```
decrypt_string(VirtualAllocSTR, 12, KEY_FOR_STRING_DECRYPTION, 16i64);
v6 = GetModuleHandleA("kernel32.dll");
VirtualAlloc = GetProcAddress(v6, VirtualAllocSTR, v7, v8);
v14 = VirtualAlloc(0i64, Size, 12288i64, 4i64);
decrypt_buffer(enced_buffer, Size, decryption_key, 16i64);
memmove(v14, enced_buffer, Size);
flOldProtect[1] = VirtualProtect(v14, Size, PAGE_EXECUTE_READ, flOldProtect);
decrypt_string(&svchost_exe_str, 11, KEY_FOR_STRING_DECRYPTION, 16i64);
```

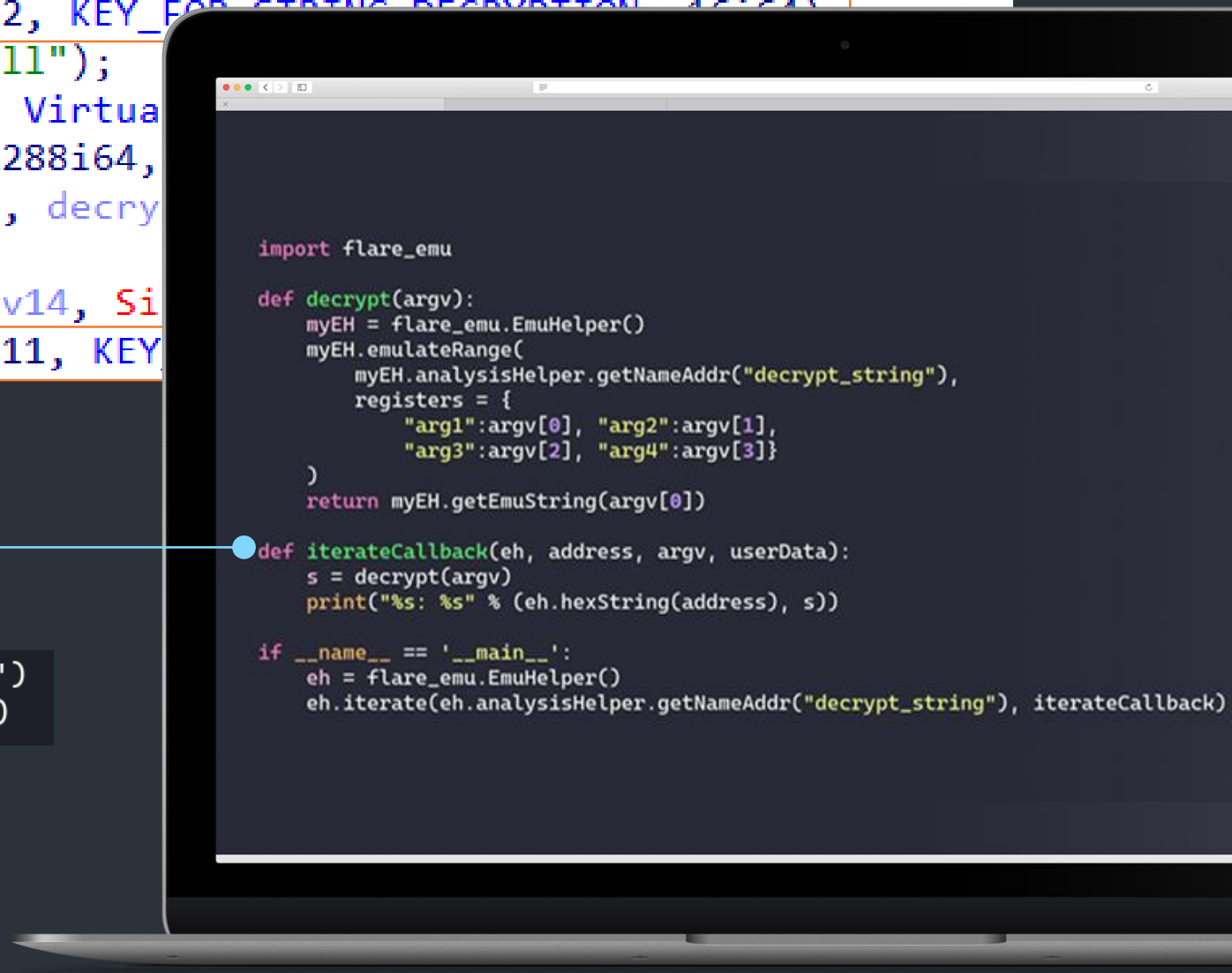
Как Flare-emu расшифровывают строки лоадера Meterpreter

SOC FORUM 2023

```
decrypt_string(VirtualAllocSTR, 12, KEY_F
v6 = GetModuleHandleA("kernel32.dll");
VirtualAlloc = GetProcAddress(v6, Virtua
v14 = VirtualAlloc(0i64, Size, 12288i64,
decrypt_buffer(enced_buffer, Size, decry
memmove(v14, enced_buffer, Size);
flOldProtect[1] = VirtualProtect(v14, Si
decrypt_string(&svchost_exe_str, 11, KEY
```

From

```
00007FF711371A55: bytearray(b'VirtualAlloc')
00007FF711371B35: bytearray(b'svchost.exe')
```



Z3

Инструмент SMT (Z3)

SOC FORUM 2023

ФУНКЦИИ

- Решение уравнений и крякмис

ОСОБЕННОСТЬ

- Более умный и быстрый метод подбора

ВПО CotxRAT

SOC FORUM 2023

1 Полнофункциональный RAT

2 Изначально был замечен в гос. секторе восточной Азии

3 Dr. Web отмечает его использование уже в Российском сегменте

ВПО CotxRAT: разбор кода

SOC FORUM 2023

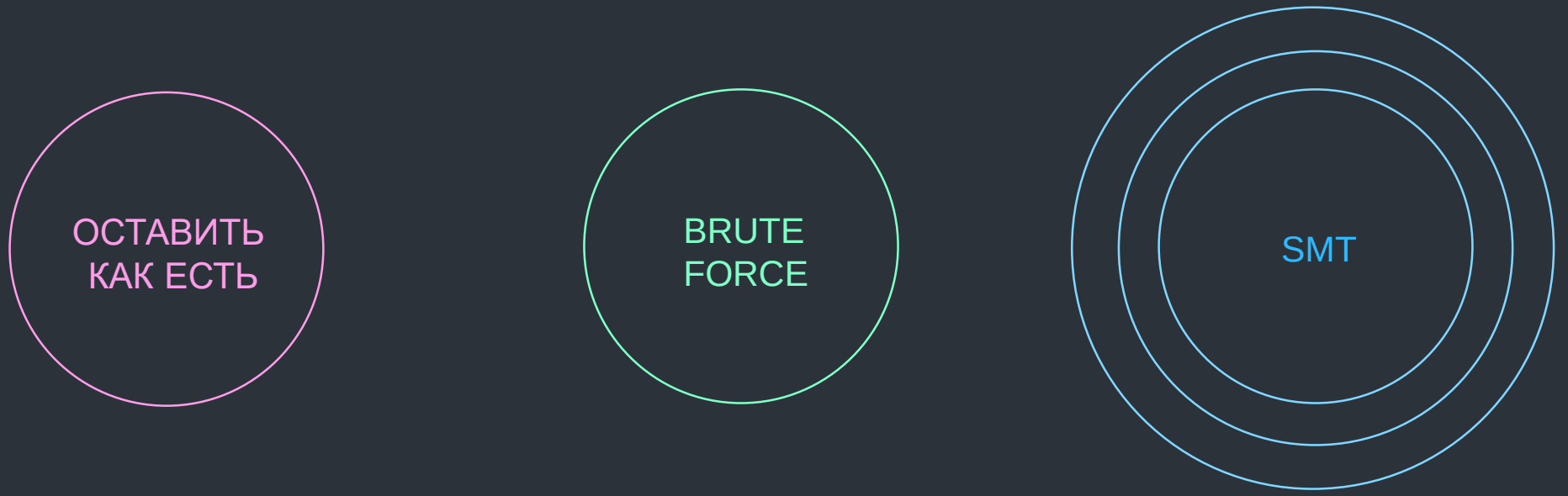
- Берется имя файла
- Оно хешируется
- Хеш сверяется с константами

Какие имена файлов?

```
v3 = filename_of_a_parent;
v4 = 0;
if ( !filename_of_a_parent[0] )
  goto LABEL_18;
do
{
  ++v3;
  v5 = v2;
  v2 = *v3;
  v4 = v5 + 17 * v4;
}
while ( *v3 );
if ( v4 != 0xA54ACF71 && v4 != 0xCD163D44 )
{
LABEL_18:
  memset(cur_exe_full_filename, 0, sizeof(cur_exe_full_filename));
  G_GetModuleFileNameA(0, cur_exe_full_filename, 2048);
```

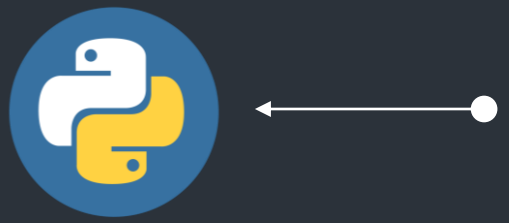
ВПО CotxRAT: варианты решения

SOC FORUM 2023



Инструмент Z3 в действии

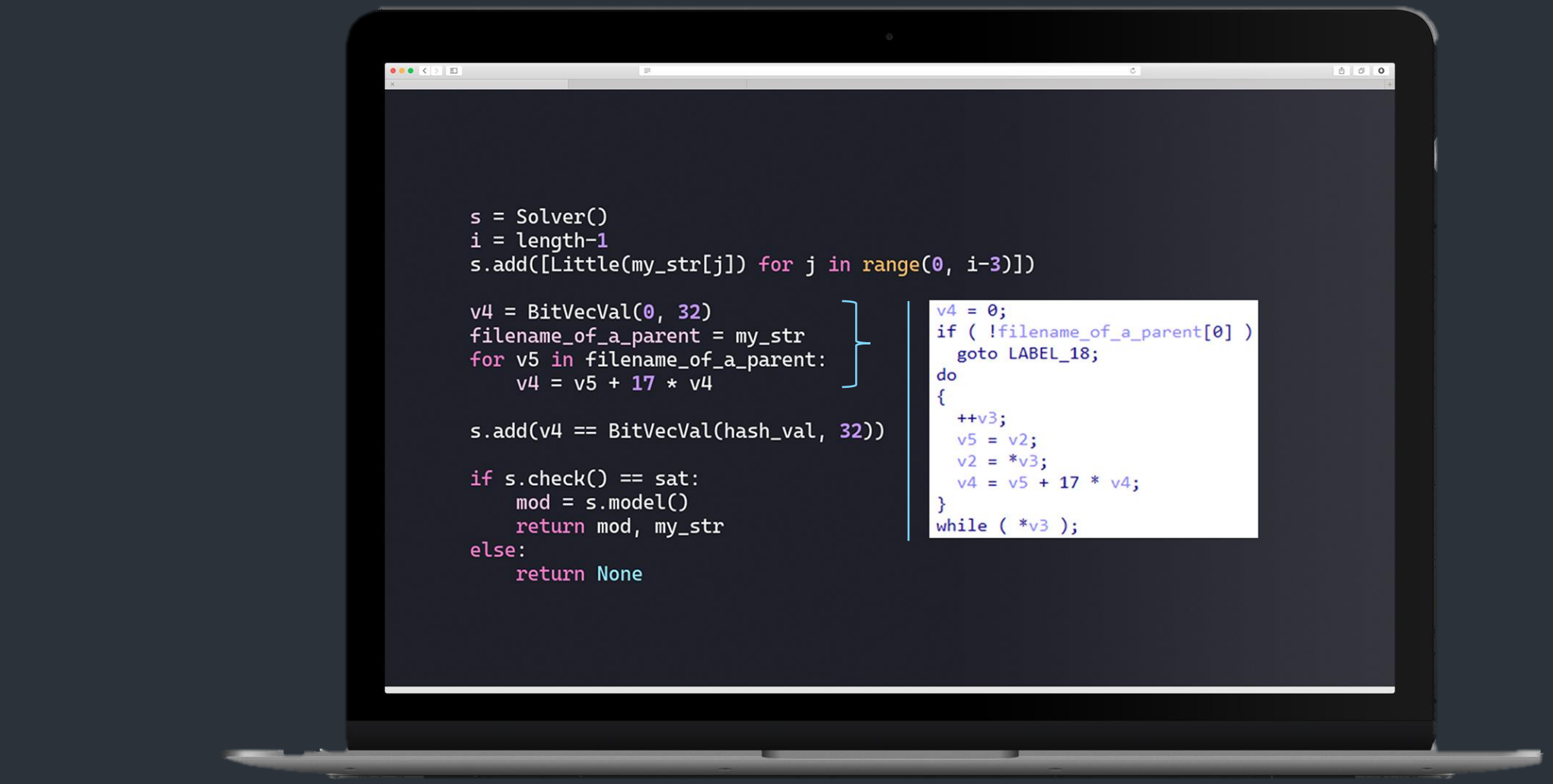
SOC FORUM 2023



```
v4 = 0;
if ( !filename_of_a_parent[0] )
  goto LABEL_18;
do
{
  ++v3;
  v5 = v2;
  v2 = *v3;
  v4 = v5 + 17 * v4;
}
while ( *v3 );
```

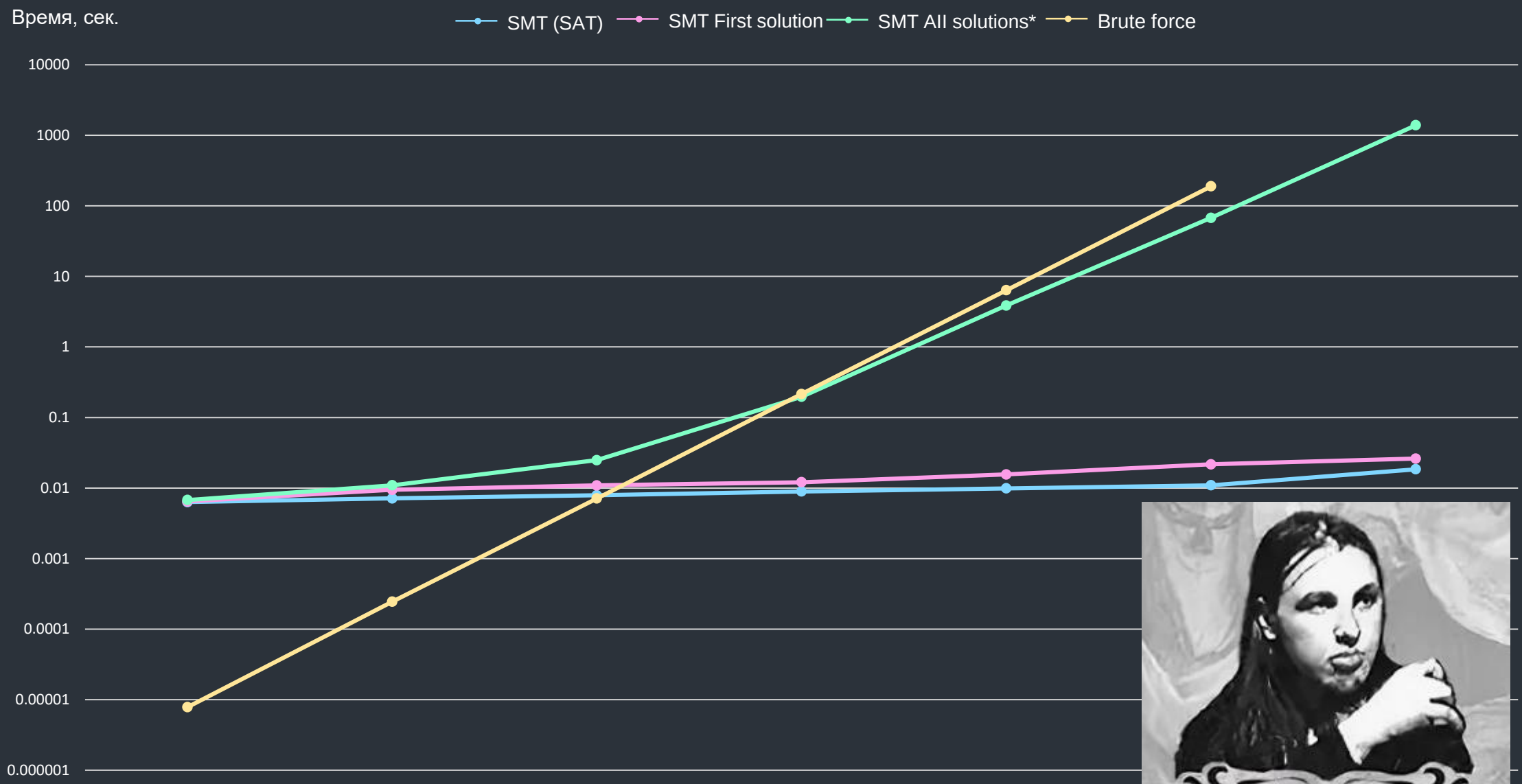
HexRays (часть CotxRAT)

Инструмент Z3 в действии



Сравнение времени подбора строки разными способами

SOC FORUM 2023



Длина подбираемой строки

Triton

Инструмент Triton

SOC FORUM 2023

ФУНКЦИИ

- Движок символьного исполнения

ОСОБЕННОСТЬ

- На C++ с байдингами для python
- NextGen эмуляции кода с возможностью строить уравнения и решать при помощи SMT-солверов – понимание, что такое символьное исполнение

Obsidium и ВПО

SOC FORUM 2023

1 Обнаружен в использовании для ВПО SgnitLoader, Arkei, RedLineStealer

2 Обнаружен нами для использования стилера

Защита ПО Obsidium

SOC FORUM 2023

КОММЕРЧЕСКИЙ ПАКЕР

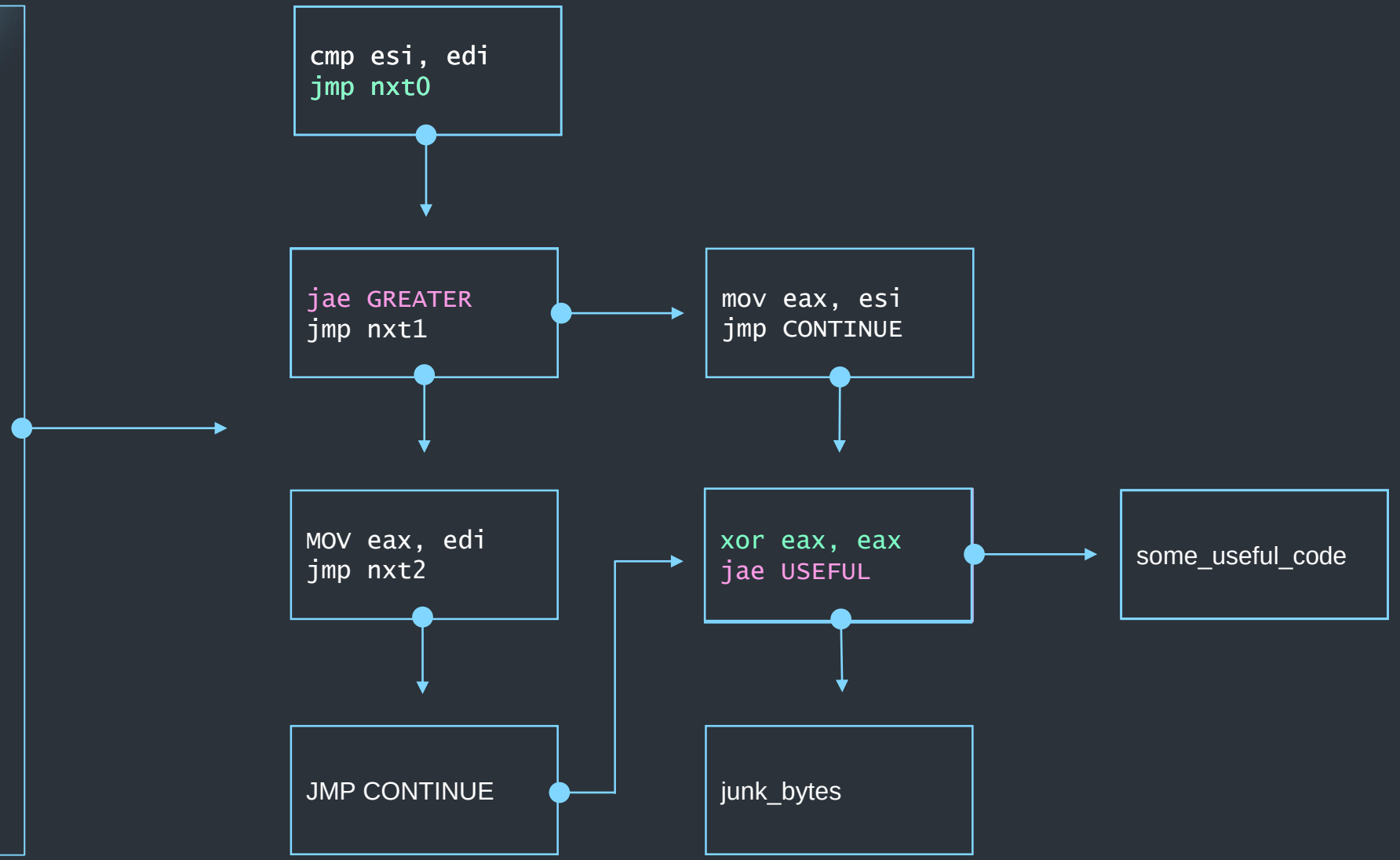
ОБНАРУЖЕН ПРИ АНАЛИЗЕ ВРЕДОНОСНОГО КОДА

ЗАЩИТА ОТ СТАТИЧЕСКОГО И ДИНАМИЧЕСКОГО АНАЛИЗА

Obsidium: убираем обфускацию

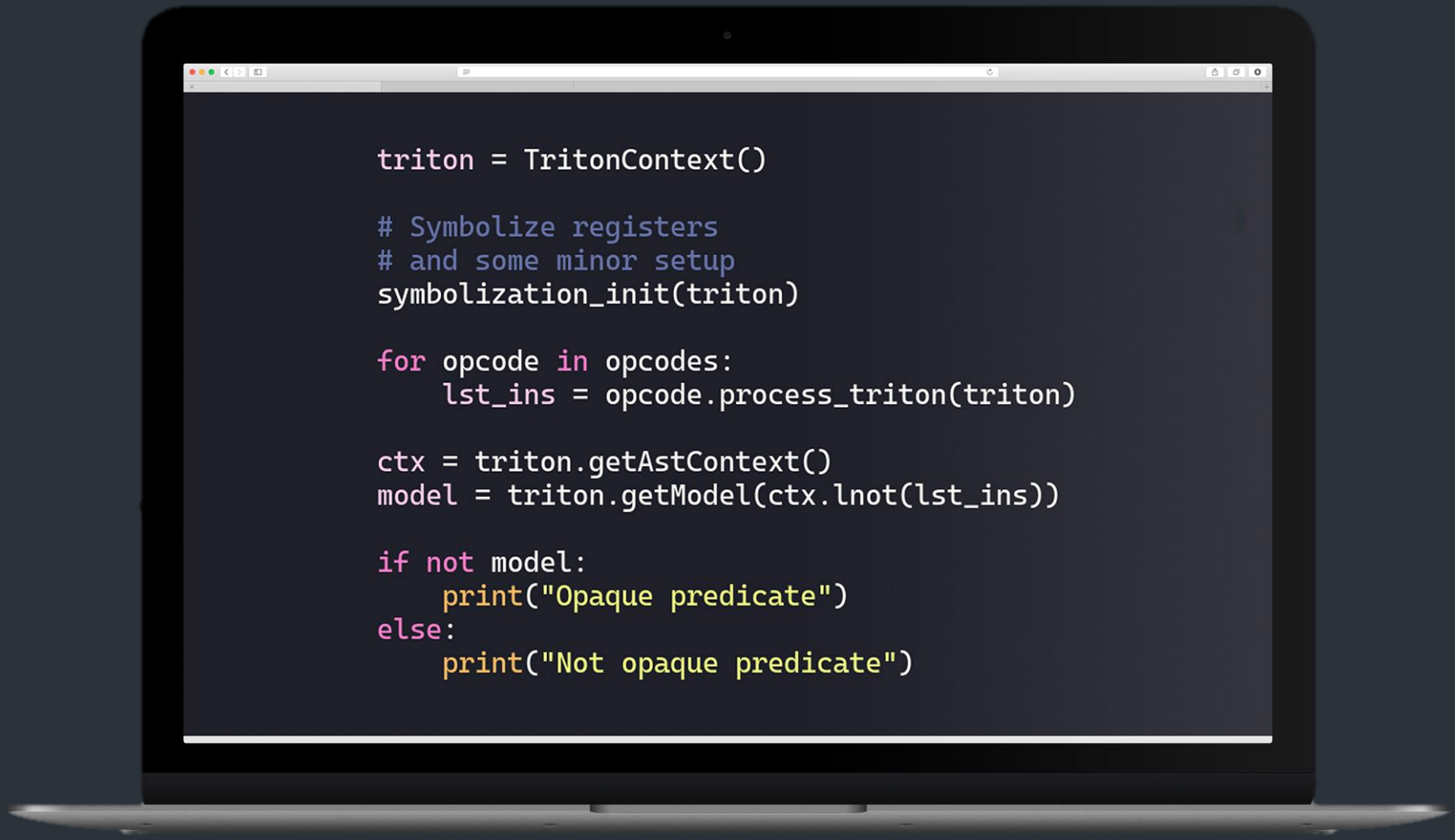
SOC FORUM 2023

- Находим условные переходы
- Берем предыдущий опкод
- Выполняем эти два опкода в Triton
- Просим Triton построить модель для последней инструкции
- DONE ☺



Инструмент Triton в действии

SOC FORUM 2023



```
triton = TritonContext()

# Symbolize registers
# and some minor setup
symbolization_init(triton)

for opcode in opcodes:
    lst_ins = opcode.process_triton(triton)

ctx = triton.getAstContext()
model = triton.getModel(ctx.lnot(lst_ins))

if not model:
    print("Opaque predicate")
else:
    print("Not opaque predicate")
```

Подведем итоги

Rizin

Emulation

SMT Solver

Symbolic execution

ВСКРЫВАЕМ ПАКЕР OBSIDIUM ДЛЯ АНАЛИЗА ВПО
ЧАСТЬ 1. ДЕОБФУКСКАЦИЯ

Узнать подробнее

SOC FORUM 2023

Владимир Нестор
v.nestor@rt-solar.ru
ГК «Солар»