A Case Study of FULLMETAL's PyArmor Usage

Otávio M.

Quem sou eu?

- Analista de Malware @ Kroll Inc.
- Autor @ deobfuscation.club
- Bacharelando em Matemática.
- Software Protection, (De)Obfuscation, Program Analysis, Compilers.

Agenda

- FULLMETAL STEALER
- PyArmor
- Patch no interpretador Python
- ELF
- Binary Ninja API

Disclaimer

G DATA
advanced analytics

https://cyber.wtf/

01

FULLMETAL

FULLMETAL: Visão Geral

Vetor de Ataque

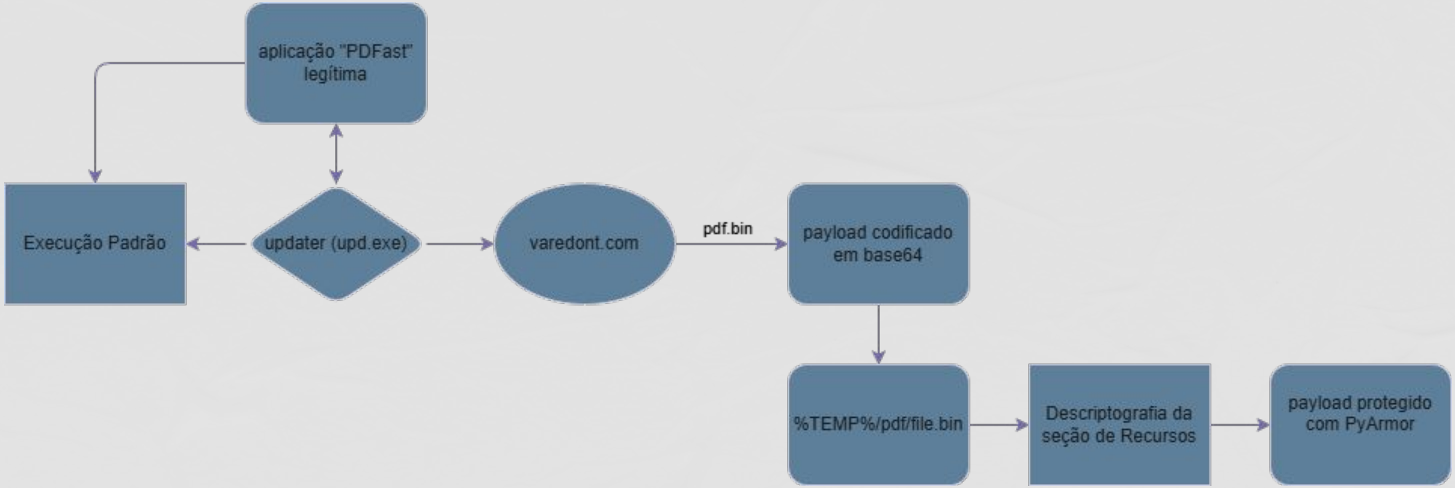
- Abuso de aplicações legítimas de manuseio de PDFs
- Comprometimento do “*updater*” da aplicação
- Segundo estágio codificado em base64
- *%TEMP%/pdf/file.bin*

Capacidades

- Stealer
- PyArmor
- Multi-arquitetura
- Detecção de ambiente virtual
- Interação com Browsers
- Interação com Cloud
- Persistencia Através de scheduled tasks

https://malpedia.caad.fkie.fraunhofer.de/details/win.fullmetal

FULLMETAL: Corrente de Infecção



aplicação "PDFast" legítima

Execução Padrão

updater (upd.exe)

varedont.com

pdf.bin

payload codificado em base64

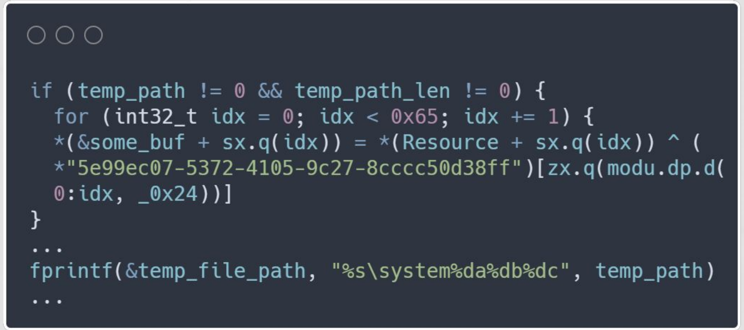
%TEMP%/pdf/file.bin

Descriptografia da seção de Recursos

payload protegido com PyArmor

FULLMETAL: Payload Final

- **Seção .rsrc contém o payload final**
    - *“CUSTOMDATA”*
- **Criptografado com uma cifra XOR.**
- **Arquivo com nome pseudo-randômico recebe o payload.**
    - Similar a *%TEMP%\system26506a16168b4007c.exe*
- **Escreve o payload final em disco.**



```
if (temp_path != 0 && temp_path_len != 0) {
  for (int32_t idx = 0; idx < 0x65; idx += 1) {
  *(&some_buf + sx.q(idx)) = *(Resource + sx.q(idx)) ^ (
  *"5e99ec07-5372-4105-9c27-8cccc50d38ff")[zx.q(modu.dp.d(
  0:idx, _0x24))]
}
...
fprintf(&temp_file_path, "%s\system%da%db%dc", temp_path)
...
```

https://www.virustotal.com/gui/file/6c2d8a47cbb42721bd7a6fc2ddcce705e9255596bdfe4b0829949bcfc262a4b1

02

PyArmor

PyArmor: O que é?

Casos de Uso

- Proteger software
- Previnir engenharia reversa
- “Vincular” o software a uma máquina específica
- Expiração ou Licenciamento de Software

Mecanismos de Proteção

- Packing
- Modo BCC
- Modo RFT
- Modo de assembler dinâmico
- Themida

PyArmor: Visão Geral

- **Packing**
  - Pyinstaller

- **Modo BCC**
  - “Transpila” código Python para código C
  - Compila para código nativo
  - Impossível de recuperar 100% o código ofuscado
  - Imports também podem ser protegidos com o modo BCC

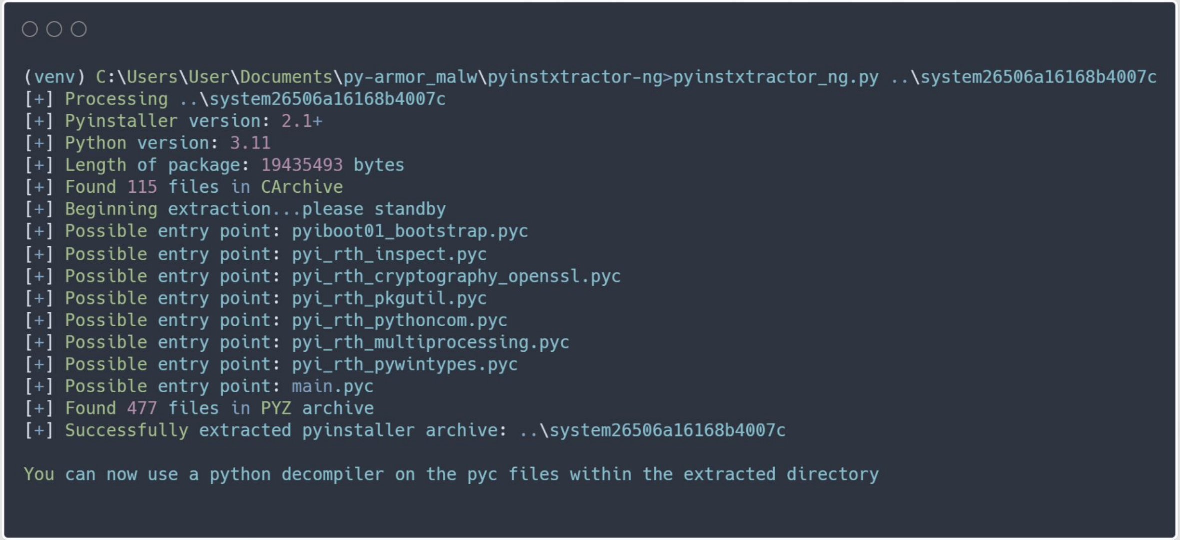
- **Modo RFT**
  - Renomeia funções, métodos, classes, variáveis, argumentos e imports para nomes aleatórios

- **Modo de assembler dinâmico**
  - GNU lightning
  - Assembla dinamicamente código x86 para computar o IV do GCM
  - Aparenta ser opcional, nem sempre aplicado
  - Pouca/Nenhuma pesquisa ou documentação até o momento

PyArmor: Packing

- **PyInstaller**
  - Software open-source que agrupa uma aplicação Python e todas as suas dependências em um único pacote.
  - Permite ao usuário rodar a aplicação sem a necessidade de ter o Python instalado.
  - pyinstxtractor-ng
  - Retorna um *.pyc

PyArmor: Packing



```
(venv) C:\Users\User\Documents\py-armor_malw\pyinstxtractor-ng>pyinstxtractor_ng.py ..\system26506a16168b4007c
[+] Processing ..\system26506a16168b4007c
[+] Pyinstaller version: 2.1+
[+] Python version: 3.11
[+] Length of package: 19435493 bytes
[+] Found 115 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: pyi_rth_cryptography_openssl.pyc
[+] Possible entry point: pyi_rth_pkgutil.pyc
[+] Possible entry point: pyi_rth_pythoncom.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: pyi_rth_pywintypes.pyc
[+] Possible entry point: main.pyc
[+] Found 477 files in PYZ archive
[+] Successfully extracted pyinstaller archive: ..\system26506a16168b4007c

You can now use a python decompiler on the pyc files within the extracted directory
```

pyinstxtractor-ng output

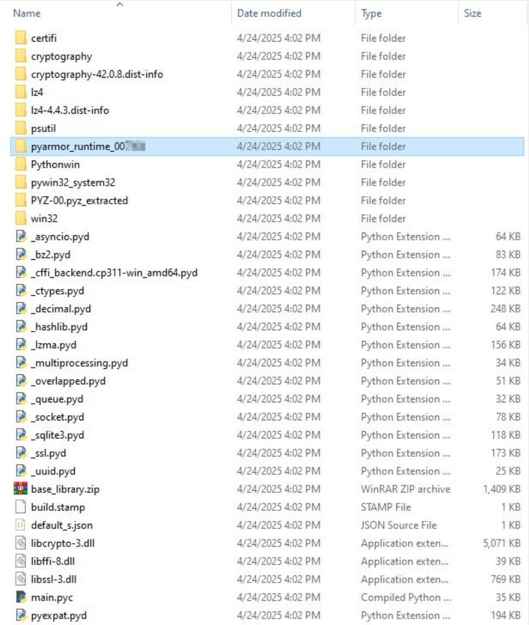
PyArmor: Estrutura de Arquivos

- **main.pyc**
  - Bytecode compilado

- **pyarmor_runtime_00XXXX/**
  - pyarmor_runtime.pyd
  - Interpretador Python modificado pelo PyArmor

- ***.pyd**
  - 64-bit DLLs

- ***.pyz**
  - Zip com header que permite ser invocado por codigo
  - 7zip

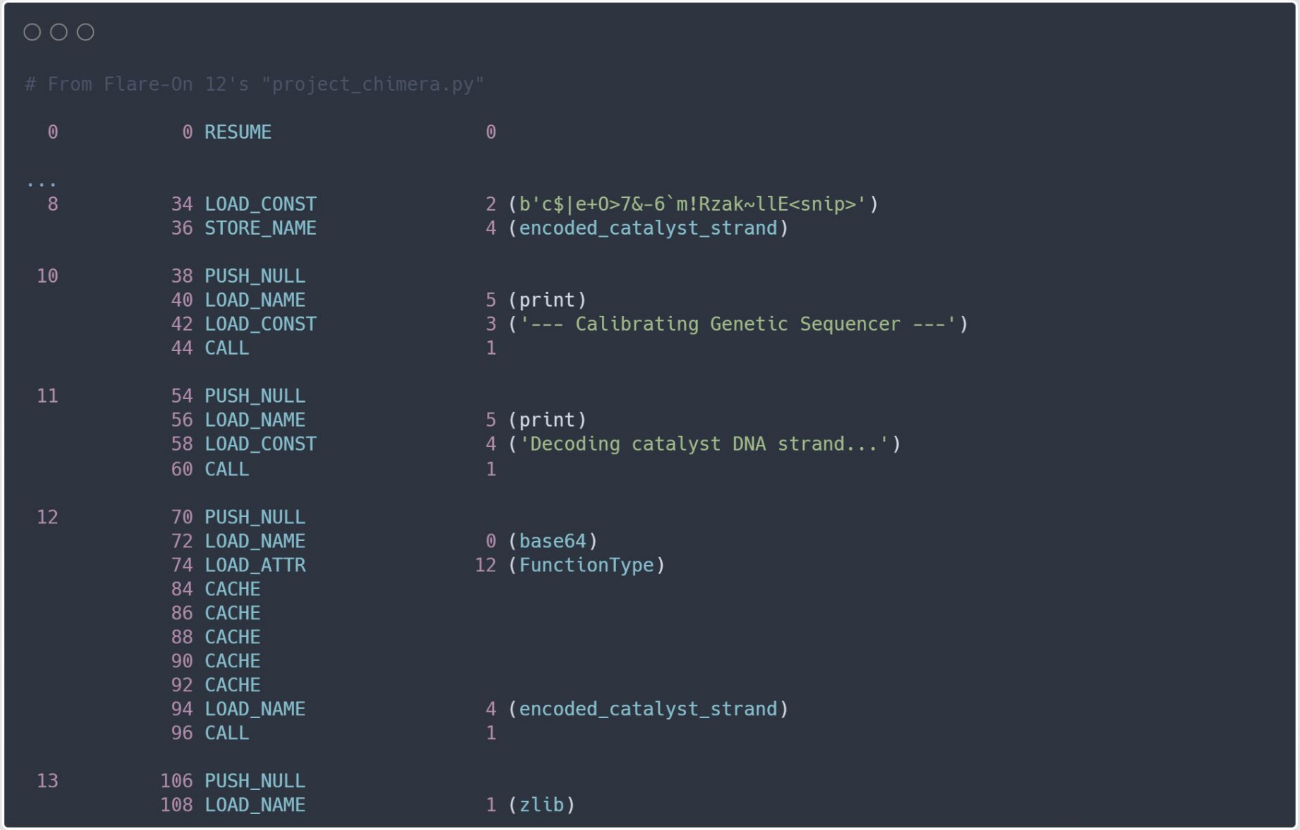


Name	Date modified	Type	Size
certifi	4/24/2025 4:02 PM	File folder	
cryptography	4/24/2025 4:02 PM	File folder	
cryptography-42.0.8.dist-info	4/24/2025 4:02 PM	File folder	
lz4	4/24/2025 4:02 PM	File folder	
lz4-4.4.3.dist-info	4/24/2025 4:02 PM	File folder	
psutil	4/24/2025 4:02 PM	File folder	
pyarmor_runtime_00[redacted]	4/24/2025 4:02 PM	File folder	
Pythonwin	4/24/2025 4:02 PM	File folder	
pywin32_system32	4/24/2025 4:02 PM	File folder	
PYZ-00.pyz_extracted	4/24/2025 4:02 PM	File folder	
win32	4/24/2025 4:02 PM	File folder	
_asyncio.pyd	4/24/2025 4:02 PM	Python Extension ...	64 KB
_bz2.pyd	4/24/2025 4:02 PM	Python Extension ...	83 KB
_cffi_backend.cp311-win_amd64.pyd	4/24/2025 4:02 PM	Python Extension ...	174 KB
_ctypes.pyd	4/24/2025 4:02 PM	Python Extension ...	122 KB
_decimal.pyd	4/24/2025 4:02 PM	Python Extension ...	248 KB
_hashlib.pyd	4/24/2025 4:02 PM	Python Extension ...	64 KB
_lzma.pyd	4/24/2025 4:02 PM	Python Extension ...	156 KB
_multiprocessing.pyd	4/24/2025 4:02 PM	Python Extension ...	34 KB
_overlapped.pyd	4/24/2025 4:02 PM	Python Extension ...	51 KB
_queue.pyd	4/24/2025 4:02 PM	Python Extension ...	32 KB
_socket.pyd	4/24/2025 4:02 PM	Python Extension ...	78 KB
_sqlite3.pyd	4/24/2025 4:02 PM	Python Extension ...	118 KB
_ssl.pyd	4/24/2025 4:02 PM	Python Extension ...	173 KB
_uuid.pyd	4/24/2025 4:02 PM	Python Extension ...	25 KB
base_library.zip	4/24/2025 4:02 PM	WinRAR ZIP archive	1,409 KB
build.stamp	4/24/2025 4:02 PM	STAMP File	1 KB
default_s.json	4/24/2025 4:02 PM	JSON Source File	1 KB
libcrypto-3.dll	4/24/2025 4:02 PM	Application exten...	5,071 KB
libffi-8.dll	4/24/2025 4:02 PM	Application exten...	39 KB
libssl-3.dll	4/24/2025 4:02 PM	Application exten...	769 KB
main.pyc	4/24/2025 4:02 PM	Compiled Python ...	35 KB
pyexpat.pyd	4/24/2025 4:02 PM	Python Extension ...	194 KB

PyArmor: Python Bytecode

- **Incompatível entre versões**
  - dis
- **Dependente do CPython**
- **Código da biblioteca “Marshal” frequentemente modificado**
- **3 dados importantes em .pyc’s**
  - Magic number de 4 bytes
  - Timestamp de 4 bytes
  - Código “marshalled”

- **Magic number muda conforme o código de marshalling**
- **Timestamp baseado no “Unix timestamp” do arquivo original que gerou o .pyc**
- **Resto do arquivo semelhante ao output de “marshal.dumps”**
- **Marshal != Pickle**



```
# From Flare-On 12's "project_chimera.py"

  0           0 RESUME                   0

...
  8          34 LOAD_CONST               2 (b'c$|e+O>7&-6`m!Rzak~llE<snip>')
             36 STORE_NAME               4 (encoded_catalyst_strand)

 10          38 PUSH_NULL
             40 LOAD_NAME                5 (print)
             42 LOAD_CONST               3 ('--- Calibrating Genetic Sequencer ---')
             44 CALL                     1

 11          54 PUSH_NULL
             56 LOAD_NAME                5 (print)
             58 LOAD_CONST               4 ('Decoding catalyst DNA strand...')
             60 CALL                     1

 12          70 PUSH_NULL
             72 LOAD_NAME                0 (base64)
             74 LOAD_ATTR               12 (FunctionType)
             84 CACHE
             86 CACHE
             88 CACHE
             90 CACHE
             92 CACHE
             94 LOAD_NAME                4 (encoded_catalyst_strand)
             96 CALL                     1

 13         106 PUSH_NULL
            108 LOAD_NAME                1 (zlib)
```

Flare-On’s 12 “project_chimera” challenge

PyArmor: Python Bytecode

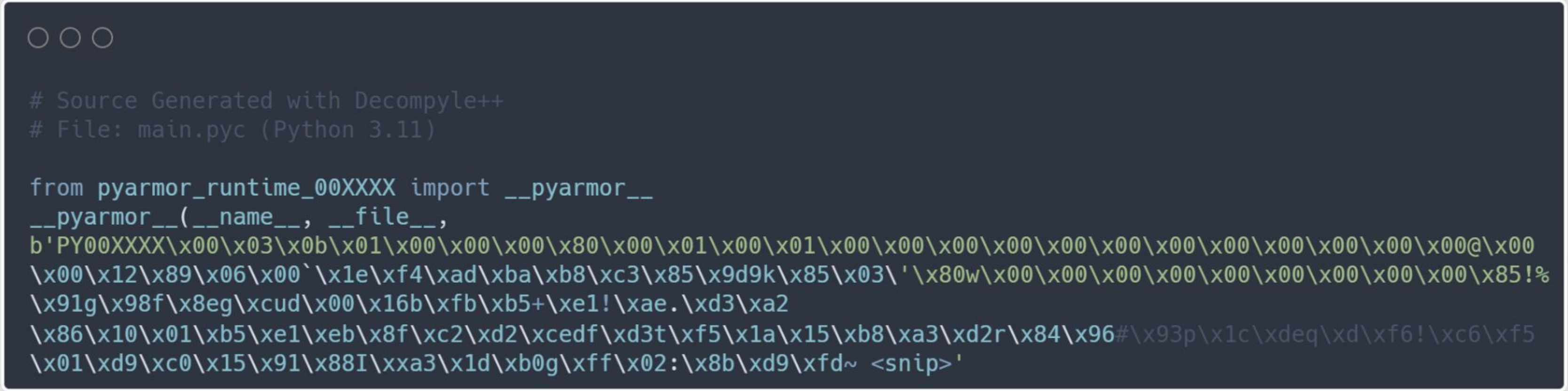
- **Alvo: 3.11**
    - Specialization
    - CACHE Instruction

- **uncompyle6 ✕**
    - <= 3.8

- **decompyle3 ✕**
    - >= 3.7

- **pycdc ✔**

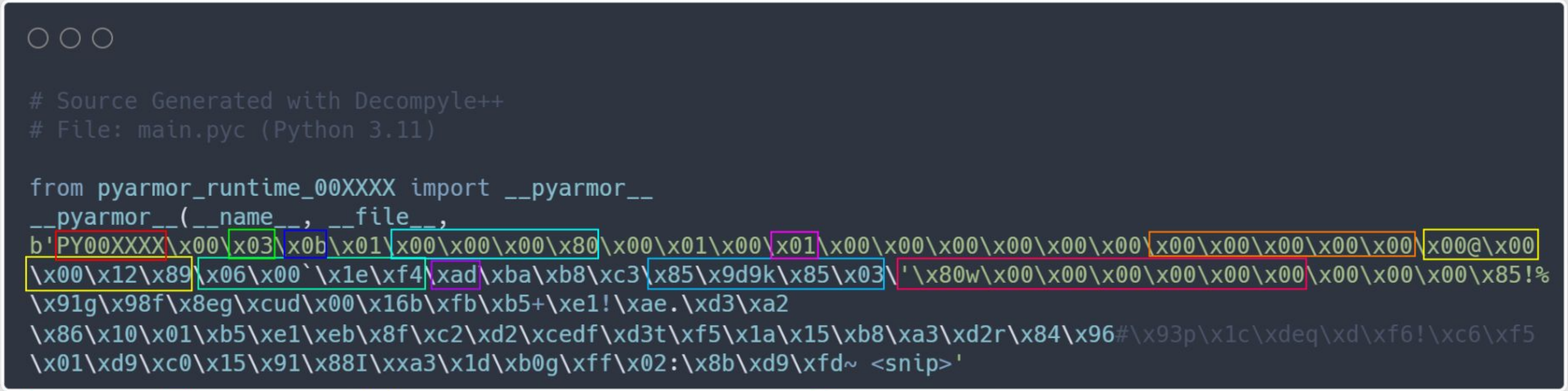
PyArmor: Descompilação



```
# Source Generated with Decompyle++
# File: main.pyc (Python 3.11)

from pyarmor_runtime_00XXXX import __pyarmor__
__pyarmor__(__name__, __file__,
b'PY00XXXX\x00\x03\x0b\x01\x00\x00\x00\x80\x00\x01\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00
\x00\x12\x89\x06\x00`\x1e\xf4\xad\xba\xb8\xc3\x85\x9d9k\x85\x03\'\x80w\x00\x00\x00\x00\x00\x00\x00\x00\x00\x85!%
\x91g\x98f\x8eg\xcud\x00\x16b\xfb\xb5+\xe1!\xae.\xd3\xa2
\x86\x10\x01\xb5\xe1\xeb\x8f\xc2\xd2\xcedf\xd3t\xf5\x1a\x15\xb8\xa3\xd2r\x84\x96#\x93p\x1c\xdeq\xd\xf6!\xc6\xf5
\x01\xd9\xc0\x15\x91\x88I\xxa3\x1d\xb0g\xff\x02:\x8b\xd9\xfd~ <snip>'
```

pycdc output

PyArmor: Descompilação



```
# Source Generated with Decompyle++
# File: main.pyc (Python 3.11)

from pyarmor_runtime_00XXXX import __pyarmor__
__pyarmor__(__name__, __file__,
b'PY00XXXX\x00\x03\x0b\x01\x00\x00\x00\x80\x00\x01\x00\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00
\x00\x12\x89\x06\x00`\x1e\xf4\xad\xba\xb8\xc3\x85\x9d9k\x85\x03\'\x80w\x00\x00\x00\x00\x00\x00\x00\x00\x00\x85!%
\x91g\x98f\x8eg\xcud\x00\x16b\xfb\xb5+\xe1!\xae.\xd3\xa2
\x86\x10\x01\xb5\xe1\xeb\x8f\xc2\xd2\xcedf\xd3t\xf5\x1a\x15\xb8\xa3\xd2r\x84\x96#\x93p\x1c\xdeq\xd\xf6!\xc6\xf5
\x01\xd9\xc0\x15\x91\x88I\xxa3\x1d\xb0g\xff\x02:\x8b\xd9\xfd~ <snip>'
```

🟥 Module Magic
🟩 Major Version
🟦 Minor Version

🟦 .pyc Magic
🟪 Protection Type
🟧 Ciphertext Offset

🟨 Ciphertext Size
🟩 IV Bytes [0:4]
🟪 GCM Applied? Any of the bits being 1: yes

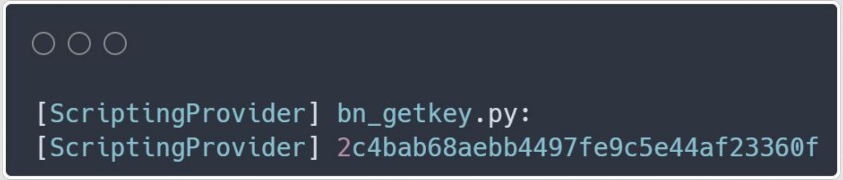
🟦 Fake IV Bytes
🟥 IV Bytes [4:12]

PyArmor: Key Derivation

- **MD5 + XOR**
    - pyarmor-vax-XXXX\x00\x00
    - RSA key

- **ida_getkey.py - GDATA**

- **bn_getkey.py**



```
[ScriptingProvider] bn_getkey.py:
[ScriptingProvider] 2c4bab68aebb4497fe9c5e44af23360f
```

bn_getkey.py output

PyArmor: Descriptografia

- **decrypt_gcm.py**
  - .dec
  - .dec.elf

- **Docker com interpretador modificado (GDATA)**
  - 3.12 ✕

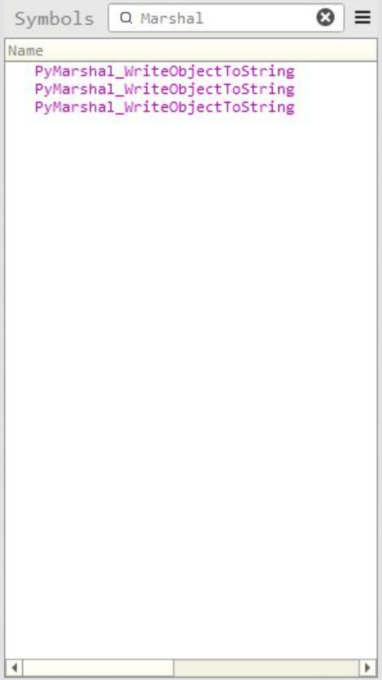
PyArmor: Marshal

- **Importa apenas *PyMarshal_WriteObjectToString***

- ***PyMarshal_ReadObjectFromString* não é importado**
    - Implementação própria

- **Específico para a versão**

- **Modifica como “code objects” são lidos**



```
PyMarshal_WriteObjectToString
PyMarshal_WriteObjectToString
PyMarshal_WriteObjectToString
```

Binary Ninja “Symbols” tab

03

3.11 PATCH

```
from pytransform import pyarmor
pyarmor(__name__, __file__, b'\x50\x59\x41\x52\x4d\x4f\x52\x00'
\x41\xf9\xa3\x0d\xb3\x55\x80\x05\x2c\x17\xc4\x40\xf8', 2)
```

```
PyObject* r_object(struct RFILE* p)
```

3.11 Patch: Mudanças

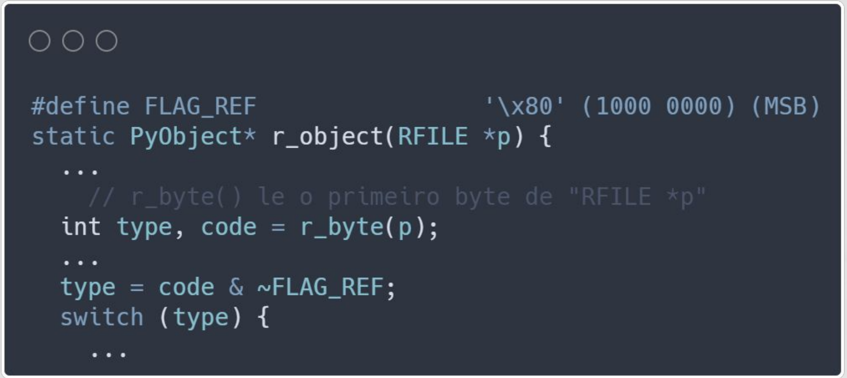
- **Specialization ✕**
- **CACHE ✕**
- **marshal.c - L1504**

3.11 Patch: r_object

- **Responsável pela “desmarshalização”**
    - Marshal format → Python Object

- **Tipo de “desmarshalização” controlado pela “*type*” flag do retorno de r_byte()**

- **Retorna PyObject ***

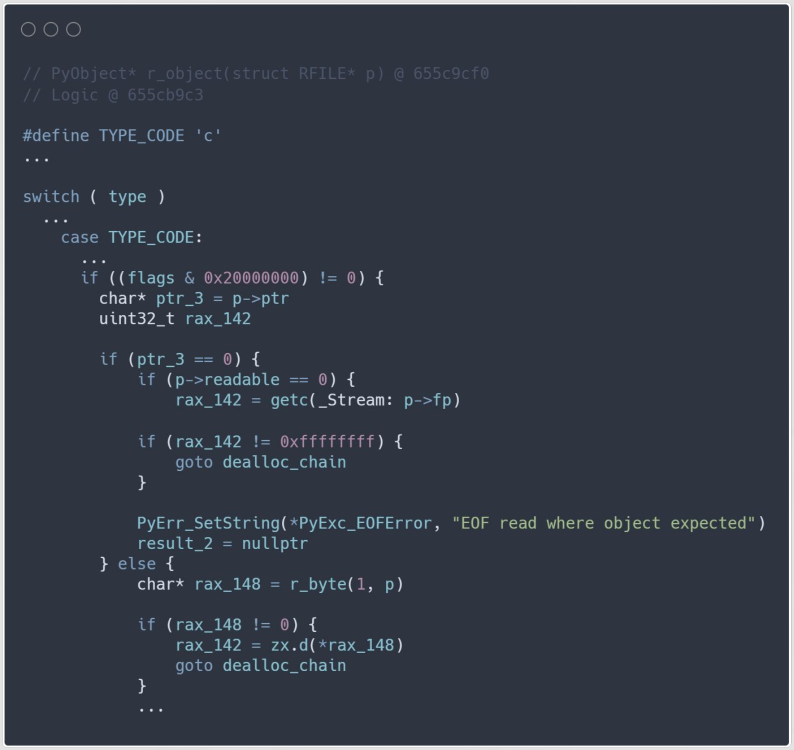


```
#define FLAG_REF                '\x80' (1000 0000) (MSB)
static PyObject* r_object(RFILE *p) {
  ...
    // r_byte() le o primeiro byte de "RFILE *p"
  int type, code = r_byte(p);
  ...
  type = code & ~FLAG_REF;
  switch (type) {
    ...
```

https://github.com/python/cpython/blob/c4ccaf4b1051b3c1ae0138a9c92657606f578fbd/Python/marshal.c#L1160

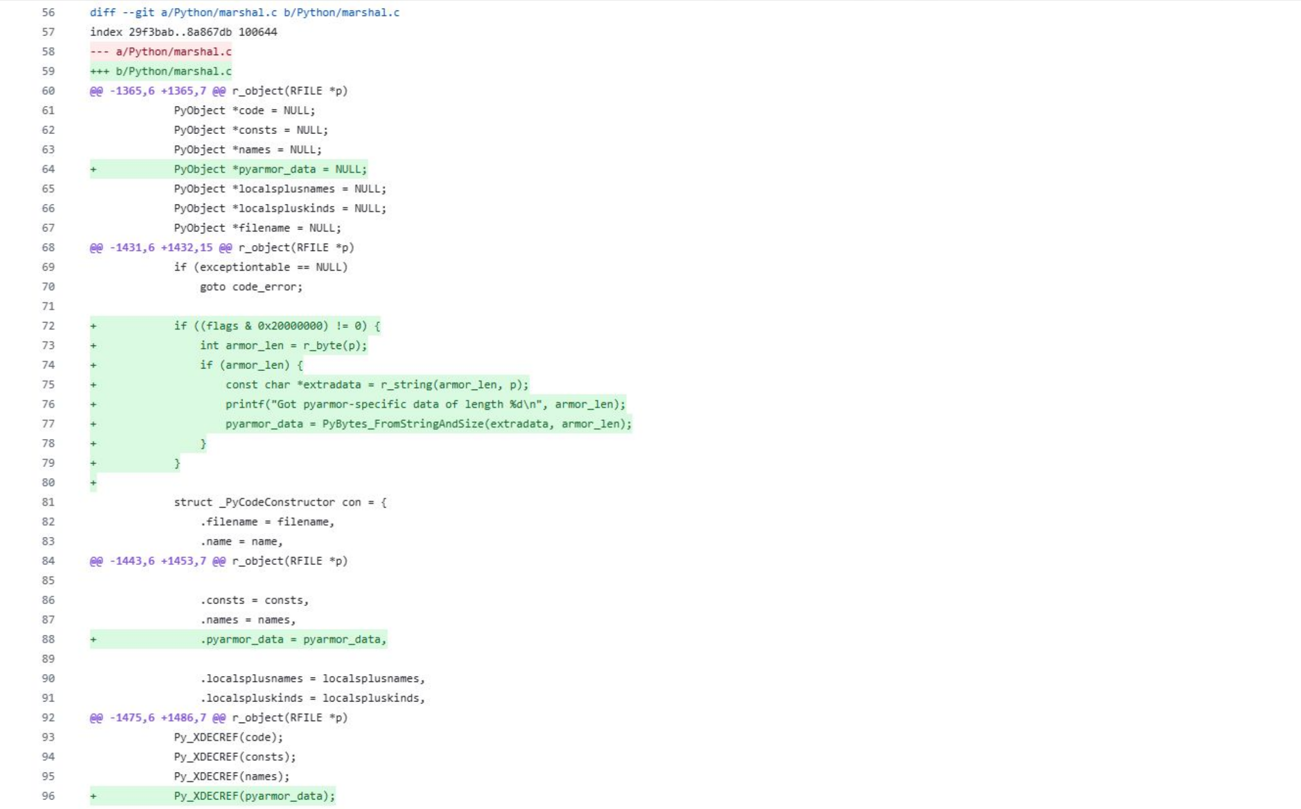
3.11 Patch: r_object

- **r_byte() inlined**
- **Bitwise AND para/com a “BCC Flag”**
  - 0x20000000
- **Bytes adicionais são lidos antes do “code object”**
  - específicos para uso do modo BCC



```
// PyObject* r_object(struct RFILE* p) @ 655c9cf0
// Logic @ 655cb9c3

#define TYPE_CODE 'c'
...

switch ( type )
  ...
    case TYPE_CODE:
      ...
      if ((flags & 0x20000000) != 0) {
        char* ptr_3 = p->ptr
        uint32_t rax_142

        if (ptr_3 == 0) {
            if (p->readable == 0) {
                rax_142 = getc(_Stream: p->fp)

            if (rax_142 != 0xffffffff) {
                goto dealloc_chain
            }

            PyErr_SetString(*PyExc_EOFError, "EOF read where object expected")
            result_2 = nullptr
        } else {
            char* rax_148 = r_byte(1, p)

            if (rax_148 != 0) {
                rax_142 = zx.d(*rax_148)
                goto dealloc_chain
            }
            ...
```

Pyarmor_runtime_.pyd.bndb @ 655cb9c3



```
diff --git a/Python/marshal.c b/Python/marshal.c
index 29f3bab..8a867db 100644
--- a/Python/marshal.c
+++ b/Python/marshal.c
@@ -1365,6 +1365,7 @@ r_object(RFILE *p)
             PyObject *code = NULL;
             PyObject *consts = NULL;
             PyObject *names = NULL;
+            PyObject *pyarmor_data = NULL;
             PyObject *localsplusnames = NULL;
             PyObject *localspluskinds = NULL;
             PyObject *filename = NULL;
@@ -1431,6 +1432,15 @@ r_object(RFILE *p)
             if (exceptiontable == NULL)
                 goto code_error;

+            if ((flags & 0x20000000) != 0) {
+                int armor_len = r_byte(p);
+                if (armor_len) {
+                    const char *extradata = r_string(armor_len, p);
+                    printf("Got pyarmor-specific data of length %d\n", armor_len);
+                    pyarmor_data = PyBytes_FromStringAndSize(extradata, armor_len);
+                }
+            }
+
             struct _PyCodeConstructor con = {
                 .filename = filename,
                 .name = name,
@@ -1443,6 +1453,7 @@ r_object(RFILE *p)

                 .consts = consts,
                 .names = names,
+                .pyarmor_data = pyarmor_data,

                 .localsplusnames = localsplusnames,
                 .localspluskinds = localspluskinds,
@@ -1475,6 +1486,7 @@ r_object(RFILE *p)
             Py_XDECREF(code);
             Py_XDECREF(consts);
             Py_XDECREF(names);
+            Py_XDECREF(pyarmor_data);
```

https://github.com/GDATAAdvancedAnalytics/Pyarmor-Tooling/blob/main/py311/armor-marshal-311.patch

3.11 Patch: Docker

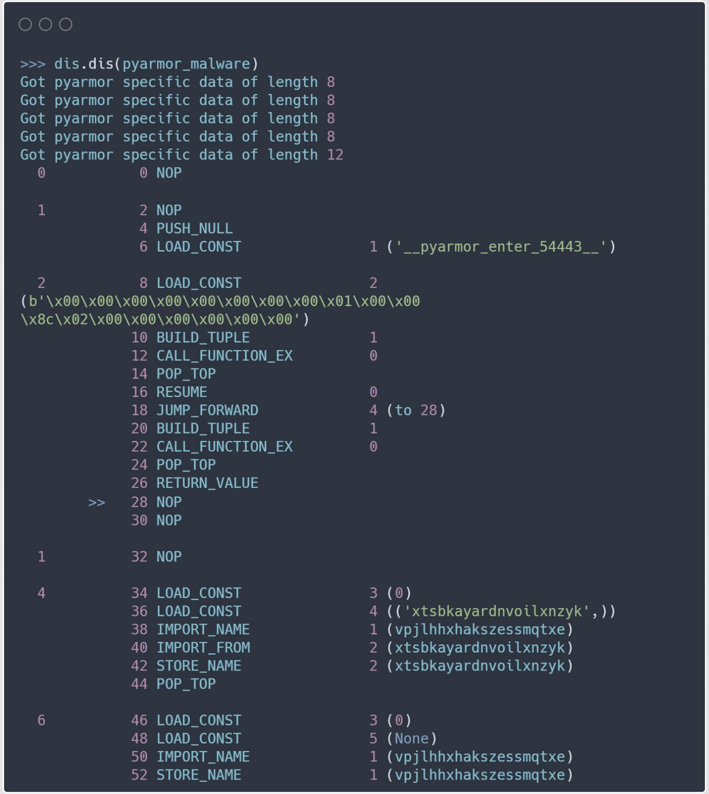
- **Interpretador patcheado**
  - git clone --branch 3.11 https://github.com/python/cpython.git
  - cd cpython
  - patch -p1 -i ../armor-marshal-311.patch
  - ./configure && make regen-all

- **Analyze_crypted_code.py**
  - Descreve como descriptografar code objects individuais
  - in *.py.dec
  - out *.py.dec2

3.11 Patch: Docker

- **decrypt_gcm.py**
  - in *.py.dec2
  - out *.py.dec2 (descriptografado)

- **disassemble.py**



```
>>> dis.dis(pyarmor_malware)
Got pyarmor specific data of length 8
Got pyarmor specific data of length 8
Got pyarmor specific data of length 8
Got pyarmor specific data of length 8
Got pyarmor specific data of length 12
  0           0 NOP

  1           2 NOP
              4 PUSH_NULL
              6 LOAD_CONST               1 ('__pyarmor_enter_54443__')

  2           8 LOAD_CONST               2
(b'\x00\x00\x00\x00\x00\x00\x00\x00\x01\x00\x00
\x8c\x02\x00\x00\x00\x00\x00\x00')
             10 BUILD_TUPLE              1
             12 CALL_FUNCTION_EX         0
             14 POP_TOP
             16 RESUME                   0
             18 JUMP_FORWARD             4 (to 28)
             20 BUILD_TUPLE              1
             22 CALL_FUNCTION_EX         0
             24 POP_TOP
             26 RETURN_VALUE
        >>   28 NOP
             30 NOP

  1          32 NOP

  4          34 LOAD_CONST               3 (0)
             36 LOAD_CONST               4 (('xtsbkayardnvoilxnzyk',))
             38 IMPORT_NAME              1 (vpjlhhxhakszessmqtxe)
             40 IMPORT_FROM              2 (xtsbkayardnvoilxnzyk)
             42 STORE_NAME               2 (xtsbkayardnvoilxnzyk)
             44 POP_TOP

  6          46 LOAD_CONST               3 (0)
             48 LOAD_CONST               5 (None)
             50 IMPORT_NAME              1 (vpjlhhxhakszessmqtxe)
             52 STORE_NAME               1 (vpjlhhxhakszessmqtxe)
```

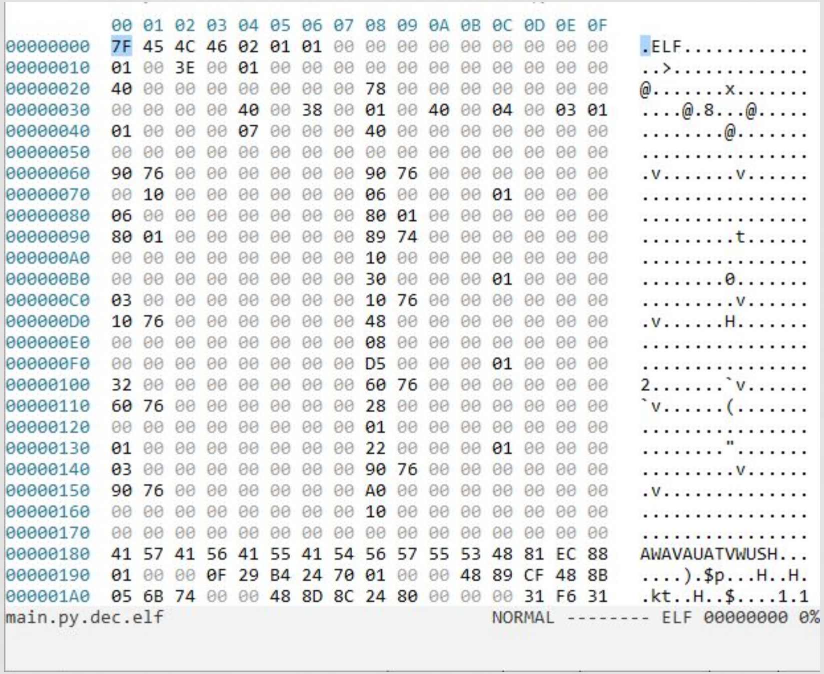
Disassembled malware bytecode

04

ELF

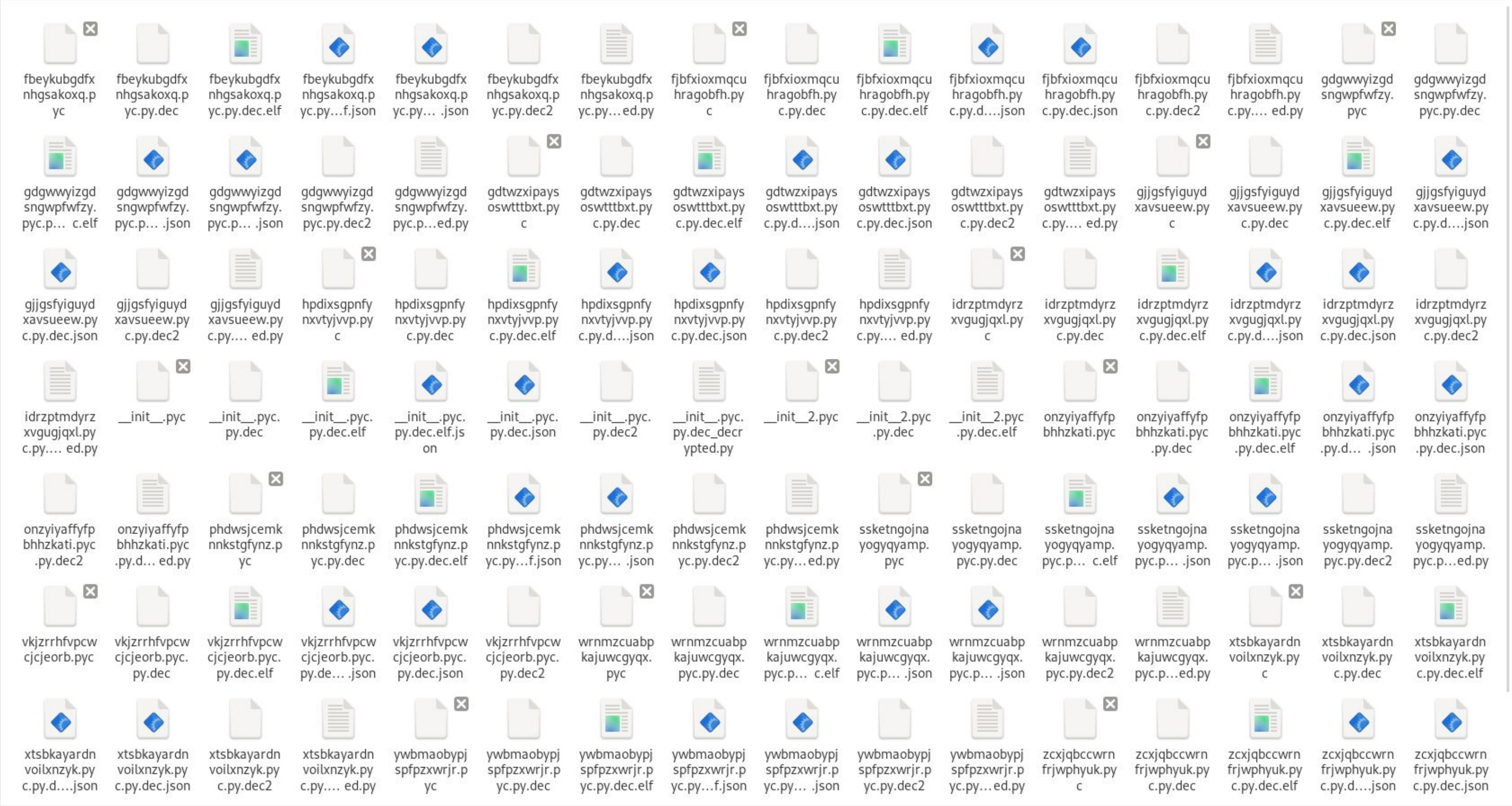
ELF: Visão Geral

- **Bytecode constantemente invoca metodos do ELF**
  - __pyarmor_bcc_54441__
  - __pyarmor_bcc_54442__
  - …
- **Maioria das capacidades maliciosas do malware**
  - Chama Imports
  - Sincroniza com a nuvem
  - Checa os argumentos do programa
- **Imports também são protegidos com o modo BCC**

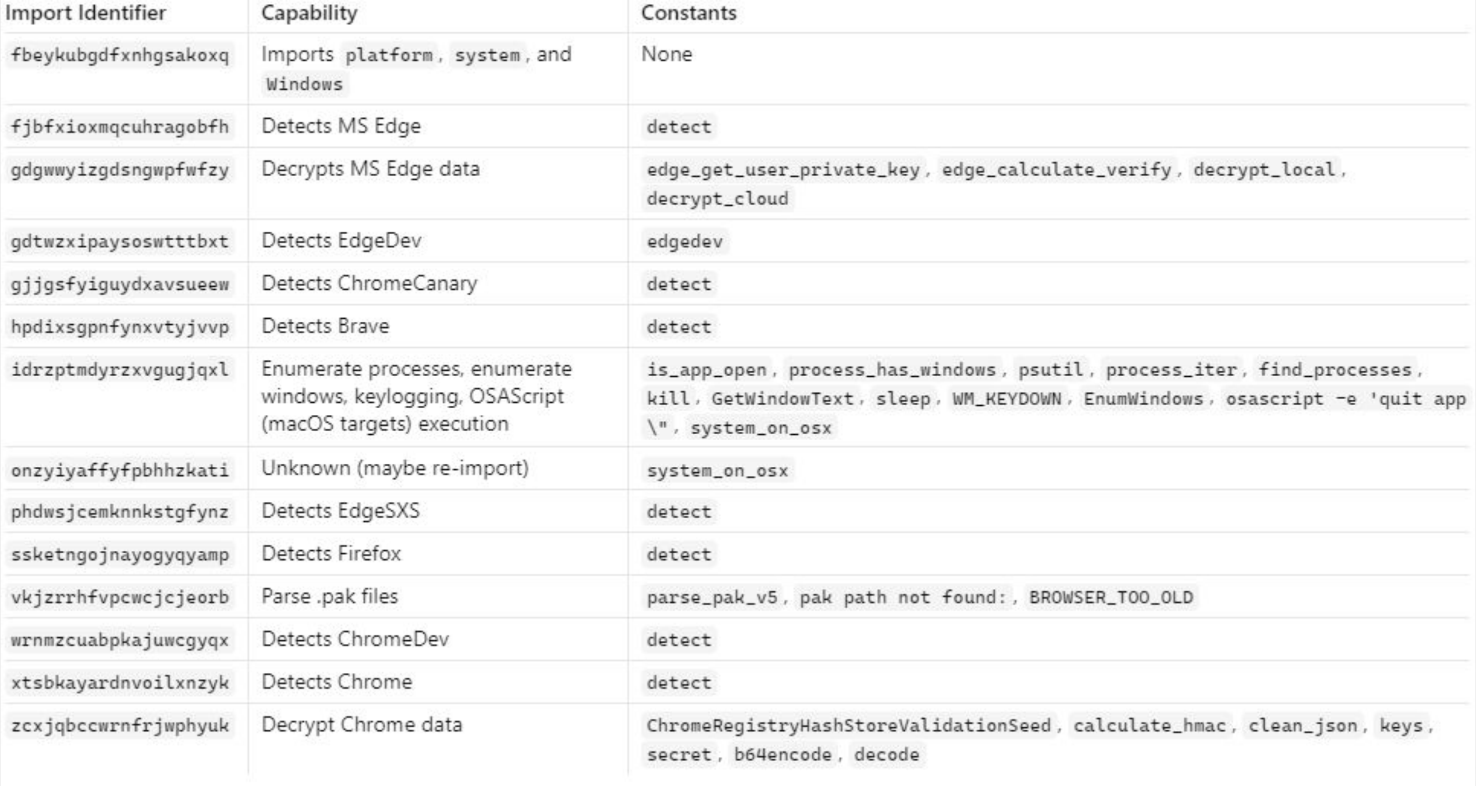


```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00   .ELF............
00000010  01 00 3E 00 01 00 00 00 00 00 00 00 00 00 00 00   ..>.............
00000020  40 00 00 00 00 00 00 00 78 00 00 00 00 00 00 00   @.......x.......
00000030  00 00 00 00 40 00 38 00 01 00 40 00 04 00 03 01   ....@.8...@.....
00000040  01 00 00 00 07 00 00 00 40 00 00 00 00 00 00 00   ........@.......
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000060  90 76 00 00 00 00 00 00 90 76 00 00 00 00 00 00   .v.......v......
00000070  00 10 00 00 00 00 00 00 06 00 00 00 01 00 00 00   ................
00000080  06 00 00 00 00 00 00 00 80 01 00 00 00 00 00 00   ................
00000090  80 01 00 00 00 00 00 00 89 74 00 00 00 00 00 00   .........t......
000000A0  00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00   ................
000000B0  00 00 00 00 00 00 00 00 30 00 00 00 01 00 00 00   ........0.......
000000C0  03 00 00 00 00 00 00 00 10 76 00 00 00 00 00 00   .........v......
000000D0  10 76 00 00 00 00 00 00 48 00 00 00 00 00 00 00   .v......H.......
000000E0  00 00 00 00 00 00 00 00 08 00 00 00 00 00 00 00   ................
000000F0  00 00 00 00 00 00 00 00 D5 00 00 00 01 00 00 00   ................
00000100  32 00 00 00 00 00 00 00 60 76 00 00 00 00 00 00   2.......`v......
00000110  60 76 00 00 00 00 00 00 28 00 00 00 00 00 00 00   `v......(.......
00000120  00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00   ................
00000130  01 00 00 00 00 00 00 00 22 00 00 00 01 00 00 00   ........".......
00000140  03 00 00 00 00 00 00 00 90 76 00 00 00 00 00 00   .........v......
00000150  90 76 00 00 00 00 00 00 A0 00 00 00 00 00 00 00   .v..............
00000160  00 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00   ................
00000170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000180  41 57 41 56 41 55 41 54 56 57 55 53 48 81 EC 88   AWAVAUATVWUSH...
00000190  01 00 00 0F 29 B4 24 70 01 00 00 48 89 CF 48 8B   ....).$p...H..H.
000001A0  05 6B 74 00 00 48 8D 8C 24 80 00 00 00 31 F6 31   .kt..H..$....1.1
main.py.dec.elf                               NORMAL -------- ELF 00000000 0%
```

1b360e4bd684c17dabea80d71888144e98407c682aac3e63d4ea0695c53966b0



```
fbeykubgdfxnhgsakoxq.pyc
fbeykubgdfxnhgsakoxq.pyc.py.dec
fbeykubgdfxnhgsakoxq.pyc.py.dec.elf
fbeykubgdfxnhgsakoxq.pyc.py…f.json
fbeykubgdfxnhgsakoxq.pyc.py… .json
fbeykubgdfxnhgsakoxq.pyc.py.dec2
fbeykubgdfxnhgsakoxq.pyc.py…ed.py
fjbfxioxmqcuhragobfh.pyc
fjbfxioxmqcuhragobfh.pyc.py.dec
fjbfxioxmqcuhragobfh.pyc.py.dec.elf
fjbfxioxmqcuhragobfh.pyc.py.d….json
fjbfxioxmqcuhragobfh.pyc.py.dec.json
fjbfxioxmqcuhragobfh.pyc.py.dec2
fjbfxioxmqcuhragobfh.pyc.py…. ed.py
gdgwwyizgdsngwpfwfzy.pyc
gdgwwyizgdsngwpfwfzy.pyc.py.dec
gdgwwyizgdsngwpfwfzy.pyc.p… c.elf
gdgwwyizgdsngwpfwfzy.pyc.p… .json
gdgwwyizgdsngwpfwfzy.pyc.p… .json
gdgwwyizgdsngwpfwfzy.pyc.py.dec2
gdgwwyizgdsngwpfwfzy.pyc.p…ed.py
gdtwzxipaysoswtttbxt.pyc
gdtwzxipaysoswtttbxt.pyc.py.dec
gdtwzxipaysoswtttbxt.pyc.py.dec.elf
gdtwzxipaysoswtttbxt.pyc.py.d….json
gdtwzxipaysoswtttbxt.pyc.py.dec.json
gdtwzxipaysoswtttbxt.pyc.py.dec2
gdtwzxipaysoswtttbxt.pyc.py…. ed.py
gjjgsfyiguydxavsueew.pyc
gjjgsfyiguydxavsueew.pyc.py.dec
gjjgsfyiguydxavsueew.pyc.py.dec.elf
gjjgsfyiguydxavsueew.pyc.py.d….json
gjjgsfyiguydxavsueew.pyc.py.dec.json
gjjgsfyiguydxavsueew.pyc.py.dec2
gjjgsfyiguydxavsueew.pyc.py…. ed.py
hpdixsgpnfynxvtyjvvp.pyc
hpdixsgpnfynxvtyjvvp.pyc.py.dec
hpdixsgpnfynxvtyjvvp.pyc.py.dec.elf
hpdixsgpnfynxvtyjvvp.pyc.py.d….json
hpdixsgpnfynxvtyjvvp.pyc.py.dec.json
hpdixsgpnfynxvtyjvvp.pyc.py.dec2
hpdixsgpnfynxvtyjvvp.pyc.py…. ed.py
idrzptmdyrzxvgugjqxl.pyc
idrzptmdyrzxvgugjqxl.pyc.py.dec
idrzptmdyrzxvgugjqxl.pyc.py.dec.elf
idrzptmdyrzxvgugjqxl.pyc.py.d….json
idrzptmdyrzxvgugjqxl.pyc.py.dec.json
idrzptmdyrzxvgugjqxl.pyc.py.dec2
idrzptmdyrzxvgugjqxl.pyc.py…. ed.py
__init__.pyc
__init__.pyc.py.dec
__init__.pyc.py.dec.elf
__init__.pyc.py.dec.elf.json
__init__.pyc.py.dec.json
__init__.pyc.py.dec2
__init__.pyc.py.dec_decrypted.py
__init__2.pyc
__init__2.pyc.py.dec
__init__2.pyc.py.dec.elf
onzyiyaffyfpbhhzkati.pyc
onzyiyaffyfpbhhzkati.pyc.py.dec
onzyiyaffyfpbhhzkati.pyc.py.dec.elf
onzyiyaffyfpbhhzkati.pyc.py.d… .json
onzyiyaffyfpbhhzkati.pyc.py.dec.json
onzyiyaffyfpbhhzkati.pyc.py.dec2
onzyiyaffyfpbhhzkati.pyc.py.d… ed.py
phdwsjcemknnkstgfynz.pyc
phdwsjcemknnkstgfynz.pyc.py.dec
phdwsjcemknnkstgfynz.pyc.py.dec.elf
phdwsjcemknnkstgfynz.pyc.py…f.json
phdwsjcemknnkstgfynz.pyc.py… .json
phdwsjcemknnkstgfynz.pyc.py.dec2
phdwsjcemknnkstgfynz.pyc.py…ed.py
ssketngojnayogyqyamp.pyc
ssketngojnayogyqyamp.pyc.py.dec
ssketngojnayogyqyamp.pyc.p… c.elf
ssketngojnayogyqyamp.pyc.p… .json
ssketngojnayogyqyamp.pyc.p… .json
ssketngojnayogyqyamp.pyc.py.dec2
ssketngojnayogyqyamp.pyc.p…ed.py
vkjzrrhfvpcwcjcjeorb.pyc
vkjzrrhfvpcwcjcjeorb.pyc.py.dec
vkjzrrhfvpcwcjcjeorb.pyc.py.dec.elf
vkjzrrhfvpcwcjcjeorb.pyc.py.de… .json
vkjzrrhfvpcwcjcjeorb.pyc.py.dec.json
vkjzrrhfvpcwcjcjeorb.pyc.py.dec2
wrnmzcuabpkajuwcgyqx.pyc
wrnmzcuabpkajuwcgyqx.pyc.py.dec
wrnmzcuabpkajuwcgyqx.pyc.p… c.elf
wrnmzcuabpkajuwcgyqx.pyc.p… .json
wrnmzcuabpkajuwcgyqx.pyc.p… .json
wrnmzcuabpkajuwcgyqx.pyc.py.dec2
wrnmzcuabpkajuwcgyqx.pyc.p…ed.py
xtsbkayardnvoilxnzyk.pyc
xtsbkayardnvoilxnzyk.pyc.py.dec
xtsbkayardnvoilxnzyk.pyc.py.dec.elf
xtsbkayardnvoilxnzyk.pyc.py.d….json
xtsbkayardnvoilxnzyk.pyc.py.dec.json
xtsbkayardnvoilxnzyk.pyc.py.dec2
xtsbkayardnvoilxnzyk.pyc.py…. ed.py
ywbmaobypjspfpzxwrjr.pyc
ywbmaobypjspfpzxwrjr.pyc.py.dec
ywbmaobypjspfpzxwrjr.pyc.py.dec.elf
ywbmaobypjspfpzxwrjr.pyc.py…f.json
ywbmaobypjspfpzxwrjr.pyc.py… .json
ywbmaobypjspfpzxwrjr.pyc.py.dec2
ywbmaobypjspfpzxwrjr.pyc.py…ed.py
zcxjqbccwrnfrjwphyuk.pyc
zcxjqbccwrnfrjwphyuk.pyc.py.dec
zcxjqbccwrnfrjwphyuk.pyc.py.dec.elf
zcxjqbccwrnfrjwphyuk.pyc.py.d….json
zcxjqbccwrnfrjwphyuk.pyc.py.dec.json
```



Import Identifier	Capability	Constants
`fbeykubgdfxnhgsakoxq`	Imports `platform`, `system`, and `Windows`	None
`fjbfxioxmqcuhragobfh`	Detects MS Edge	`detect`
`gdgwwyizgdsngwpfwfzy`	Decrypts MS Edge data	`edge_get_user_private_key`, `edge_calculate_verify`, `decrypt_local`, `decrypt_cloud`
`gdtwzxipaysoswtttbxt`	Detects EdgeDev	`edgedev`
`gjjgsfyiguydxavsueew`	Detects ChromeCanary	`detect`
`hpdixsgpnfynxvtyjvvp`	Detects Brave	`detect`
`idrzptmdyrzxvgugjqxl`	Enumerate processes, enumerate windows, keylogging, OSAScript (macOS targets) execution	`is_app_open`, `process_has_windows`, `psutil`, `process_iter`, `find_processes`, `kill`, `GetWindowText`, `sleep`, `WM_KEYDOWN`, `EnumWindows`, `osascript -e 'quit app \"`, `system_on_osx`
`onzyiyaffyfpbhhzkati`	Unknown (maybe re-import)	`system_on_osx`
`phdwsjcemknnkstgfynz`	Detects EdgeSXS	`detect`
`ssketngojnayogyqyamp`	Detects Firefox	`detect`
`vkjzrrhfvpcwcjcjeorb`	Parse .pak files	`parse_pak_v5`, `pak path not found:`, `BROWSER_TOO_OLD`
`wrnmzcuabpkajuwcgyqx`	Detects ChromeDev	`detect`
`xtsbkayardnvoilxnzyk`	Detects Chrome	`detect`
`zcxjqbccwrnfrjwphyuk`	Decrypt Chrome data	`ChromeRegistryHashStoreValidationSeed`, `calculate_hmac`, `clean_json`, `keys`, `secret`, `b64encode`, `decode`

ELF: BCC

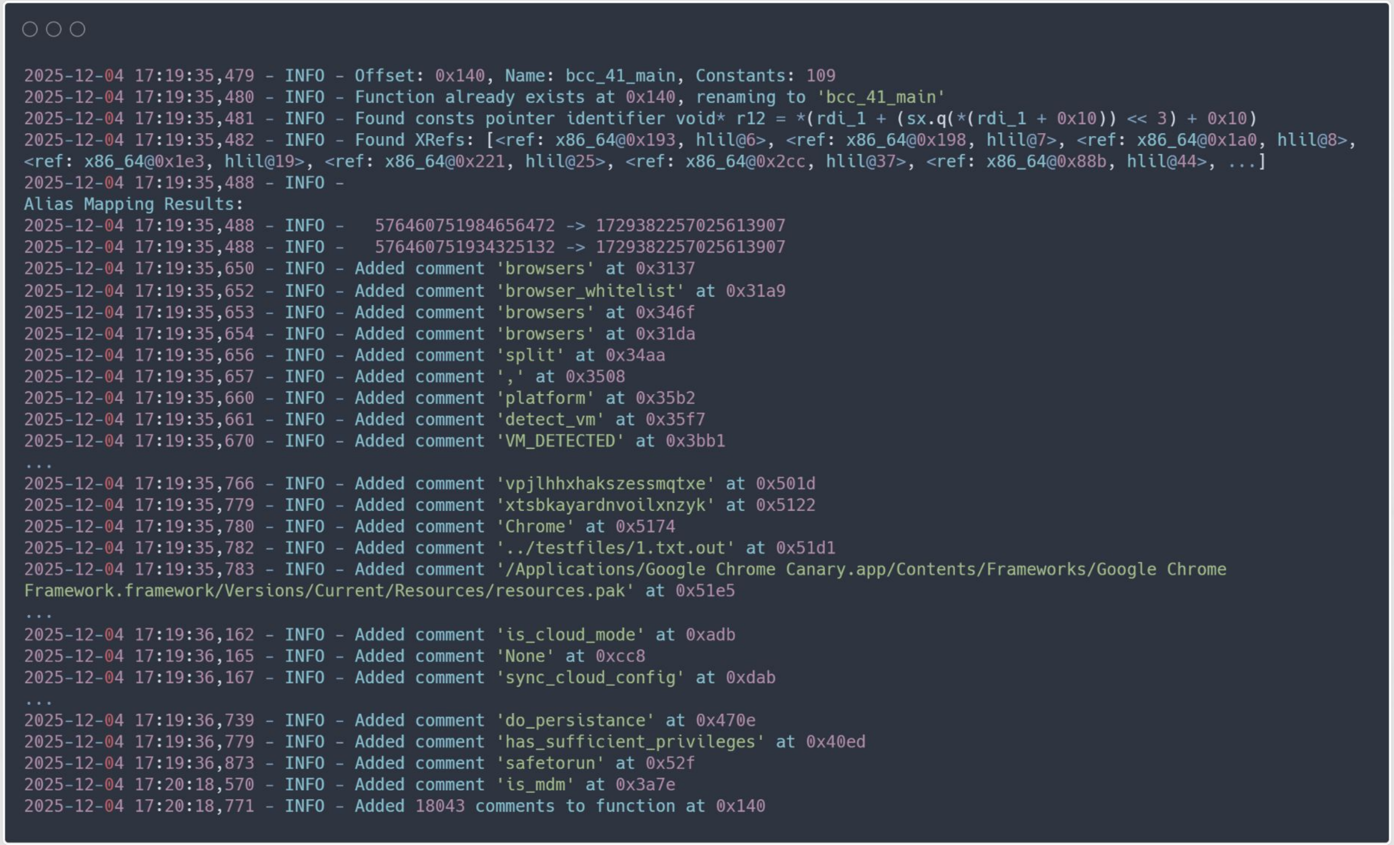
- **Memória *RWX* alocada em tempo de execução**
  - *VirtualAlloc*
  - Tamanho do ELF
- **Mapeamento de como patchear o objeto de código sendo lido**
  - *co_consts*
  - Metodos nativos injetados
- **Constante patcheada torna-se PyCMethod**
- **PyCMethod(PyObject *self) ✕**
- **PyCMethod(co->co_consts) ✔**
  - *(None, '__pyarmor_bcc_54440__', ('sys', 'exit'))*

ELF: Ferramentas BCC

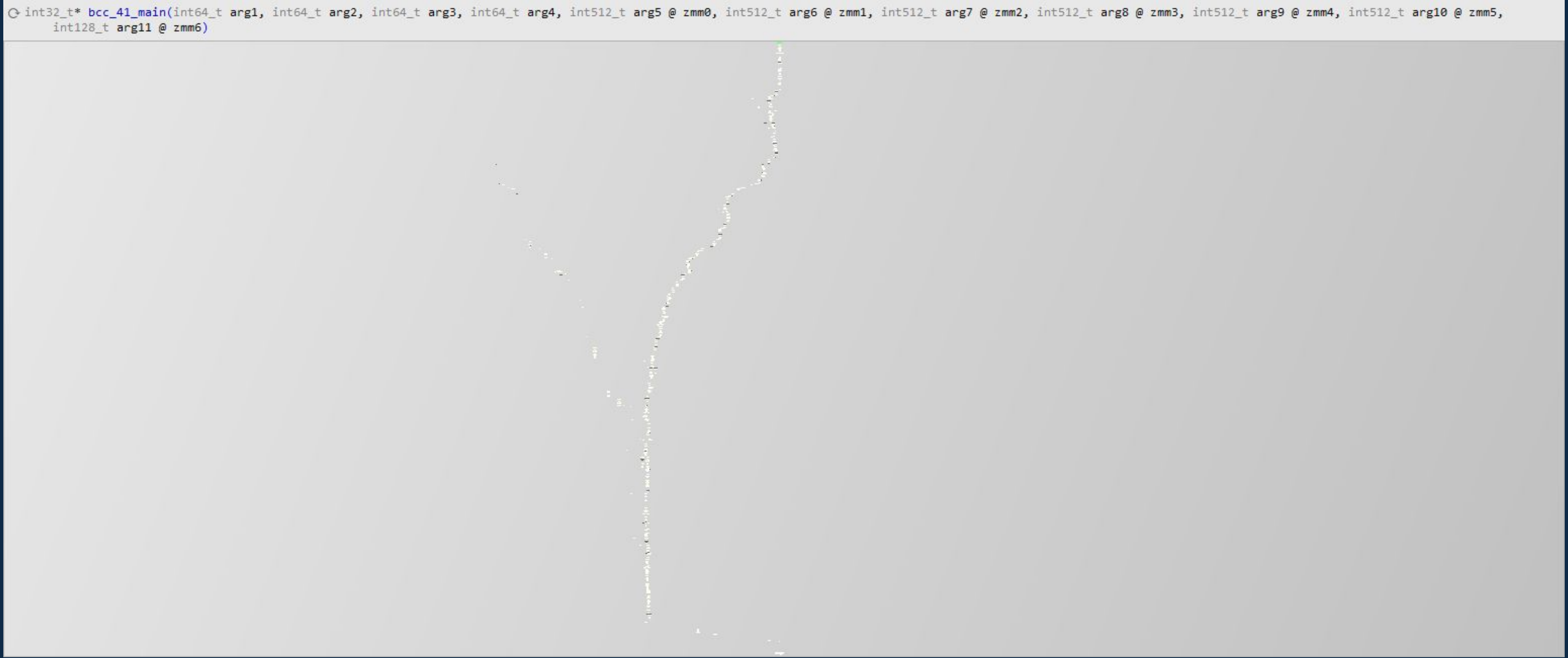
- ***Bcc_info.py***
  - *.elf.json

- **IDA-centricas ✕**

- **Binary Ninja ✔**
  - Lê o *JSON* (offsets, nomes, consts)
  - Acha o ponteiro para constantes
    - *r12 = *(arg1 + (sx.q(*(arg1 + 0x10)) << 3) + 0x10)*
  - Acha “aliases” ao ponteiro (*r12*)
  - Percorre a AST “HLIL”
  - Mapeia Offset → Index da constante → Adiciona o comentário



```
2025-12-04 17:19:35,479 - INFO - Offset: 0x140, Name: bcc_41_main, Constants: 109
2025-12-04 17:19:35,480 - INFO - Function already exists at 0x140, renaming to 'bcc_41_main'
2025-12-04 17:19:35,481 - INFO - Found consts pointer identifier void* r12 = *(rdi_1 + (sx.q(*(rdi_1 + 0x10)) << 3) + 0x10)
2025-12-04 17:19:35,482 - INFO - Found XRefs: [<ref: x86_64@0x193, hlil@6>, <ref: x86_64@0x198, hlil@7>, <ref: x86_64@0x1a0, hlil@8>,
<ref: x86_64@0x1e3, hlil@19>, <ref: x86_64@0x221, hlil@25>, <ref: x86_64@0x2cc, hlil@37>, <ref: x86_64@0x88b, hlil@44>, ...]
2025-12-04 17:19:35,488 - INFO -
Alias Mapping Results:
2025-12-04 17:19:35,488 - INFO -   576460751984656472 -> 1729382257025613907
2025-12-04 17:19:35,488 - INFO -   576460751934325132 -> 1729382257025613907
2025-12-04 17:19:35,650 - INFO - Added comment 'browsers' at 0x3137
2025-12-04 17:19:35,652 - INFO - Added comment 'browser_whitelist' at 0x31a9
2025-12-04 17:19:35,653 - INFO - Added comment 'browsers' at 0x346f
2025-12-04 17:19:35,654 - INFO - Added comment 'browsers' at 0x31da
2025-12-04 17:19:35,656 - INFO - Added comment 'split' at 0x34aa
2025-12-04 17:19:35,657 - INFO - Added comment ',' at 0x3508
2025-12-04 17:19:35,660 - INFO - Added comment 'platform' at 0x35b2
2025-12-04 17:19:35,661 - INFO - Added comment 'detect_vm' at 0x35f7
2025-12-04 17:19:35,670 - INFO - Added comment 'VM_DETECTED' at 0x3bb1
...
2025-12-04 17:19:35,766 - INFO - Added comment 'vpjlhhxhakszessmqtxe' at 0x501d
2025-12-04 17:19:35,779 - INFO - Added comment 'xtsbkayardnvoilxnzyk' at 0x5122
2025-12-04 17:19:35,780 - INFO - Added comment 'Chrome' at 0x5174
2025-12-04 17:19:35,782 - INFO - Added comment '../testfiles/1.txt.out' at 0x51d1
2025-12-04 17:19:35,783 - INFO - Added comment '/Applications/Google Chrome Canary.app/Contents/Frameworks/Google Chrome
Framework.framework/Versions/Current/Resources/resources.pak' at 0x51e5
...
2025-12-04 17:19:36,162 - INFO - Added comment 'is_cloud_mode' at 0xadb
2025-12-04 17:19:36,165 - INFO - Added comment 'None' at 0xcc8
2025-12-04 17:19:36,167 - INFO - Added comment 'sync_cloud_config' at 0xdab
...
2025-12-04 17:19:36,739 - INFO - Added comment 'do_persistance' at 0x470e
2025-12-04 17:19:36,779 - INFO - Added comment 'has_sufficient_privileges' at 0x40ed
2025-12-04 17:19:36,873 - INFO - Added comment 'safetorun' at 0x52f
2025-12-04 17:20:18,570 - INFO - Added comment 'is_mdm' at 0x3a7e
2025-12-04 17:20:18,771 - INFO - Added 18043 comments to function at 0x140
```



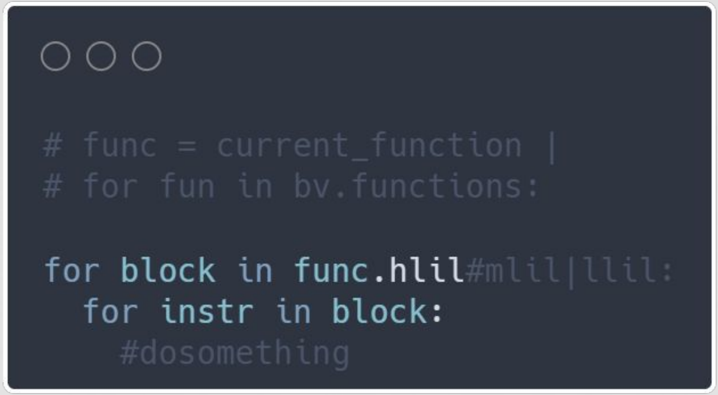
```
int32_t* bcc_41_main(int64_t arg1, int64_t arg2, int64_t arg3, int64_t arg4, int512_t arg5 @ zmm0, int512_t arg6 @ zmm1, int512_t arg7 @ zmm2, int512_t arg8 @ zmm3, int512_t arg9 @ zmm4, int512_t arg10 @ zmm5,
    int128_t arg11 @ zmm6)
```

ELF’s “bcc_41_main” subroutine

05

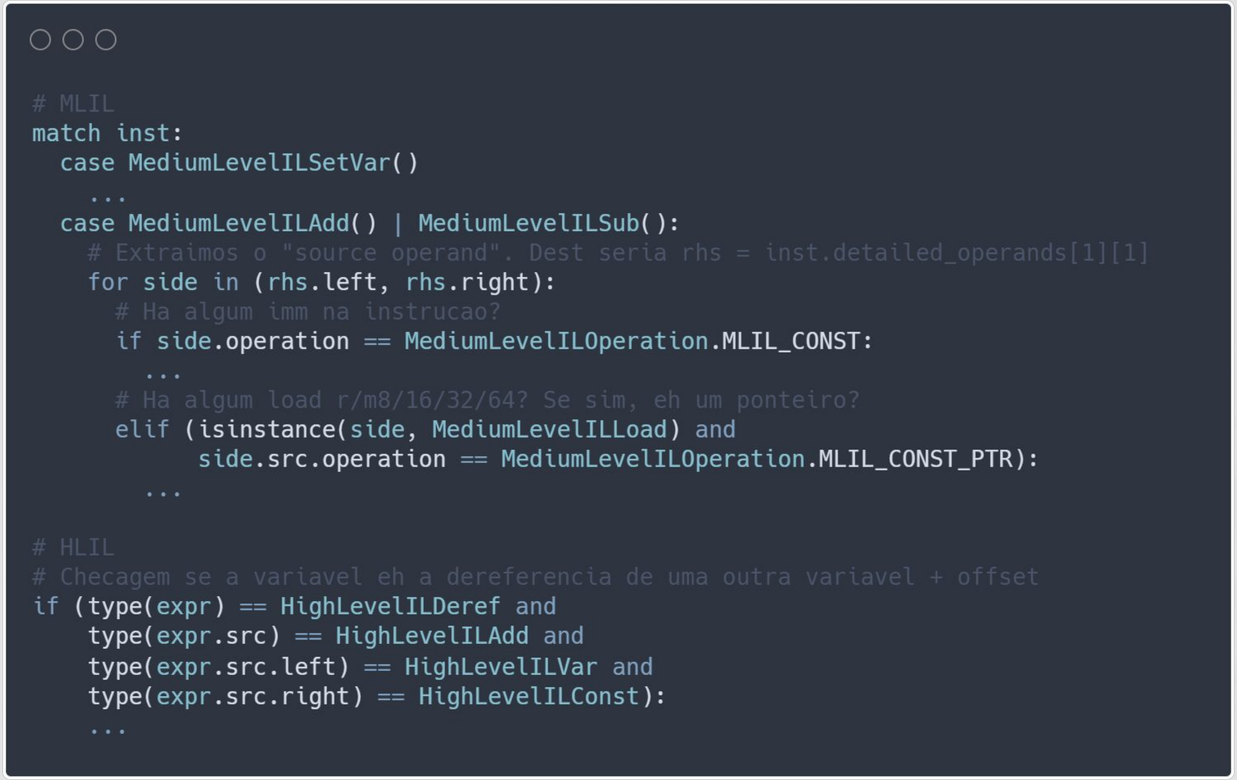
Binary Ninja API

Binary Ninja API: Iterar as sobre as instruções de um bloco



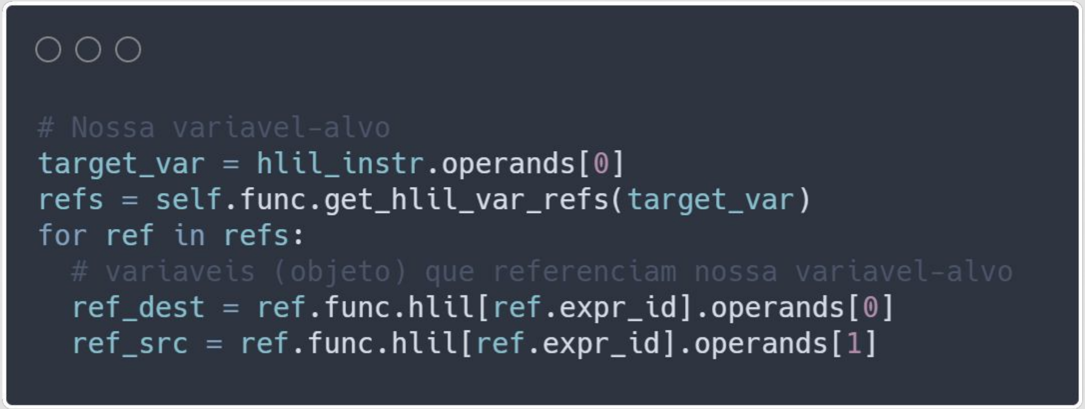
```
# func = current_function |
# for fun in bv.functions:

for block in func.hlil#mlil|llil:
  for instr in block:
    #dosomething
```

Binary Ninja API: Pattern Matching



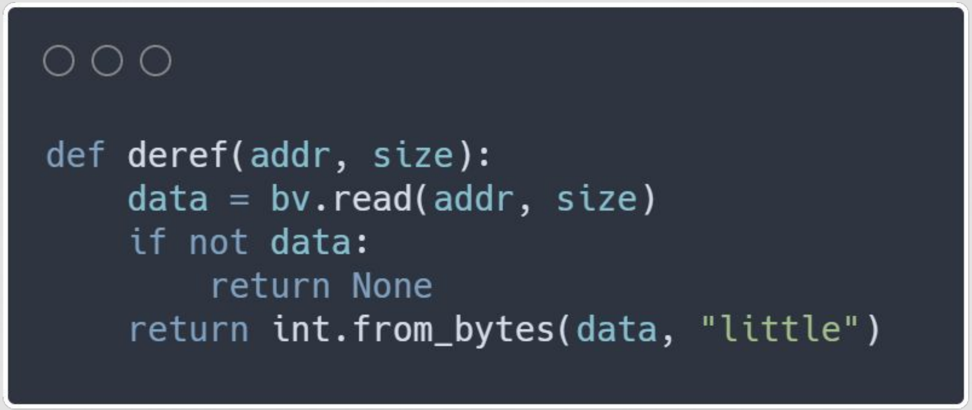
```
# MLIL
match inst:
  case MediumLevelILSetVar()
    ...
  case MediumLevelILAdd() | MediumLevelILSub():
    # Extraimos o "source operand". Dest seria rhs = inst.detailed_operands[1][1]
    for side in (rhs.left, rhs.right):
      # Ha algum imm na instrucao?
      if side.operation == MediumLevelILOperation.MLIL_CONST:
        ...
      # Ha algum load r/m8/16/32/64? Se sim, eh um ponteiro?
      elif (isinstance(side, MediumLevelILLoad) and
            side.src.operation == MediumLevelILOperation.MLIL_CONST_PTR):
        ...

# HLIL
# Checagem se a variavel eh a dereferencia de uma outra variavel + offset
if (type(expr) == HighLevelILDeref and
    type(expr.src) == HighLevelILAdd and
    type(expr.src.left) == HighLevelILVar and
    type(expr.src.right) == HighLevelILConst):
    ...
```

Binary Ninja API: XREFs de uma variável



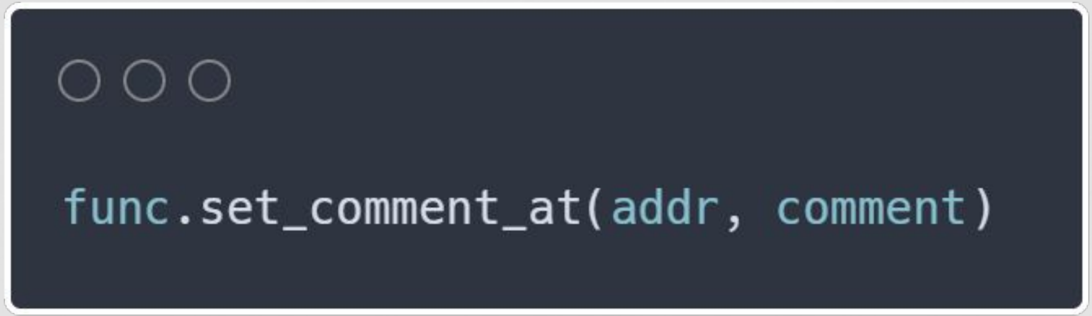
```python
# Nossa variavel-alvo
target_var = hlil_instr.operands[0]
refs = self.func.get_hlil_var_refs(target_var)
for ref in refs:
  # variaveis (objeto) que referenciam nossa variavel-alvo
  ref_dest = ref.func.hlil[ref.expr_id].operands[0]
  ref_src = ref.func.hlil[ref.expr_id].operands[1]
```

Binary Ninja API: Dereferenciar um endereço



```
def deref(addr, size):
    data = bv.read(addr, size)
    if not data:
        return None
    return int.from_bytes(data, "little")
```

Binary Ninja API: Comentar em um endereço



```
func.set_comment_at(addr, comment)
```

Binary Ninja API: Snippets

https://gist.github.com/psifertex/6fbc7532f536775194edd26290892ef7

Obrigado!

github.com/estr3llas

in/otavio8664

@radare2