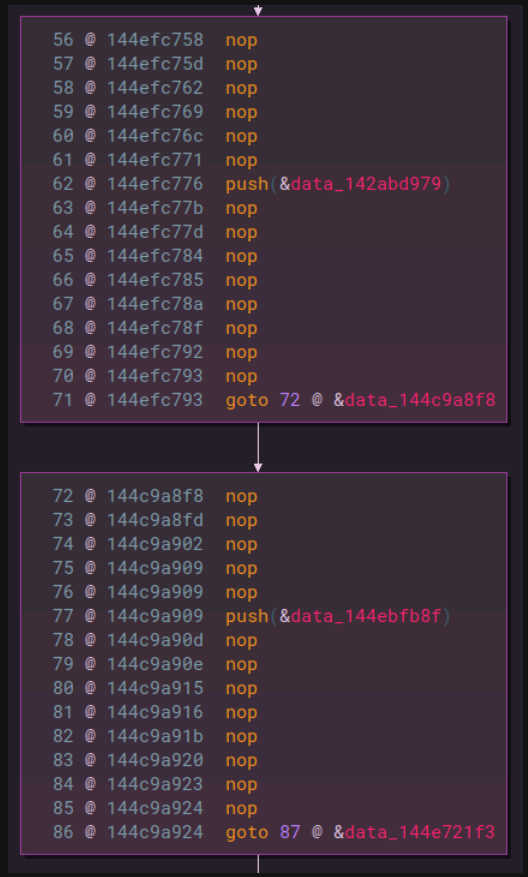
Reversing Age of Empires 2

Upgrading tools from Dark Age to Castle Age

Teaser


56 @ 144efc758 [rsp - 8 {var_f8_1}].q = r10
57 @ 144efc75d rsp = rsp - 8
58 @ 144efc762 r10 = &data_142abd979
59 @ 144efc769 push([rsp {var_f8_1}].q)
60 @ 144efc76c [rsp + 8 {var_f8_2}].q = r10
61 @ 144efc771 rsp = rsp + 8
62 @ 144efc776 r10 = [rsp - 8 {var_100_3}].q
63 @ 144efc77b push(r12)
64 @ 144efc77d r12 = &data_144c9a8f8
65 @ 144efc784 push(rax)
66 @ 144efc785 rax = [rsp + 8 {var_100_4}].q
67 @ 144efc78a [rsp + 8 {var_100_4}].q = r12
68 @ 144efc78f r12 = rax
69 @ 144efc792 rax = pop
70 @ 144efc793 temp0.q = pop
71 @ 144efc793 jump(temp0.q => 72 @ &data_144c9a8f8)
72 @ 144c9a8f8 [rsp - 8 {var_100}].q = r14
73 @ 144c9a8fd rsp = rsp - 8
74 @ 144c9a902 r14 = &data_144ebfb8f
75 @ 144c9a909 temp0.q = [rsp {var_100}].q
76 @ 144c9a909 [rsp {var_100_1}].q = r14
77 @ 144c9a909 r14 = temp0.q
78 @ 144c9a90d push(rbx)
79 @ 144c9a90e rbx = &data_144e721f3
80 @ 144c9a915 push(rax)
81 @ 144c9a916 rax = [rsp + 8 {var_108_1}].q
82 @ 144c9a91b [rsp + 8 {var_108_1}].q = rbx
83 @ 144c9a920 rbx = rax
84 @ 144c9a923 rax = pop
85 @ 144c9a924 temp0.q = pop
86 @ 144c9a924 jump(temp0.q => 87 @ &data_144e721f3)


56 @ 144efc758 nop
57 @ 144efc75d nop
58 @ 144efc762 nop
59 @ 144efc769 nop
60 @ 144efc76c nop
61 @ 144efc771 nop
62 @ 144efc776 push(&data_142abd979)
63 @ 144efc77b nop
64 @ 144efc77d nop
65 @ 144efc784 nop
66 @ 144efc785 nop
67 @ 144efc78a nop
68 @ 144efc78f nop
69 @ 144efc792 nop
70 @ 144efc793 nop
71 @ 144efc793 goto 72 @ &data_144c9a8f8
72 @ 144c9a8f8 nop
73 @ 144c9a8fd nop
74 @ 144c9a902 nop
75 @ 144c9a909 nop
76 @ 144c9a909 nop
77 @ 144c9a909 push(&data_144ebfb8f)
78 @ 144c9a90d nop
79 @ 144c9a90e nop
80 @ 144c9a915 nop
81 @ 144c9a916 nop
82 @ 144c9a91b nop
83 @ 144c9a920 nop
84 @ 144c9a923 nop
85 @ 144c9a924 nop
86 @ 144c9a924 goto 87 @ &data_144e721f3

Disclaimer

- Not affiliated with Microsoft
- Project for fun and learning

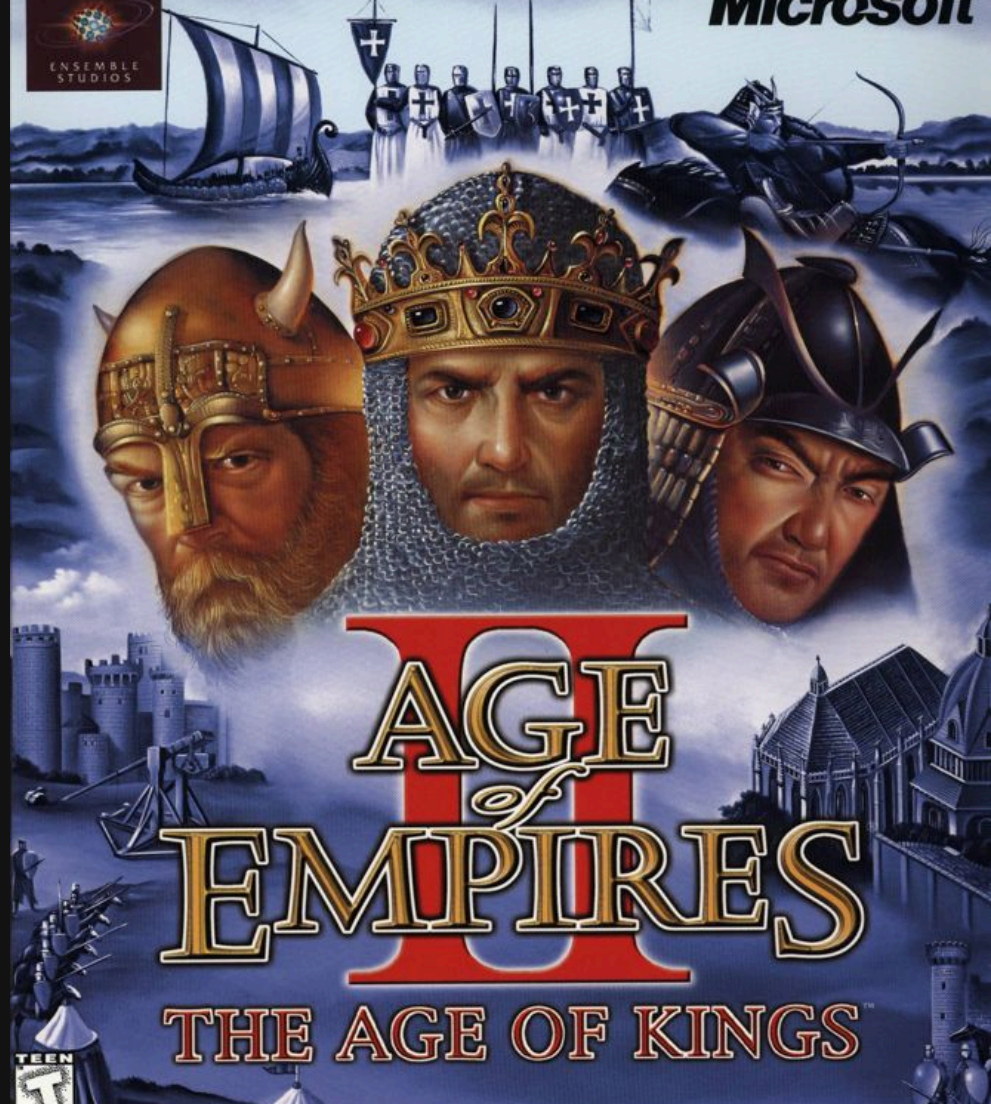
Bio

- Carl "ZetaTwo" Svensson, 34
- Swedish Londoner
- Security Engineer in finance
- CTF veteran
  - Org: Google CTF, LiveCTF, ...
  - Player: HackingForSoju, NorseCode, ...
- Publications:
  - Blog / YouTube
  - PagedOut

Age of Empires 2

Background

- RTS PC Game
- Released 1999
  - Ensemble Studios (Microsoft)
- HD Remake 2013
  - Hidden Path Entertainment
- Definitive Edition 2019
  - Forgotten Empires et al
  - ~100 patches since
- I played on/off throughout my life
  - Came back to the game with DE
  - ~1000h played
  - Top ~20% competitive



Dark Age

Manual exploration



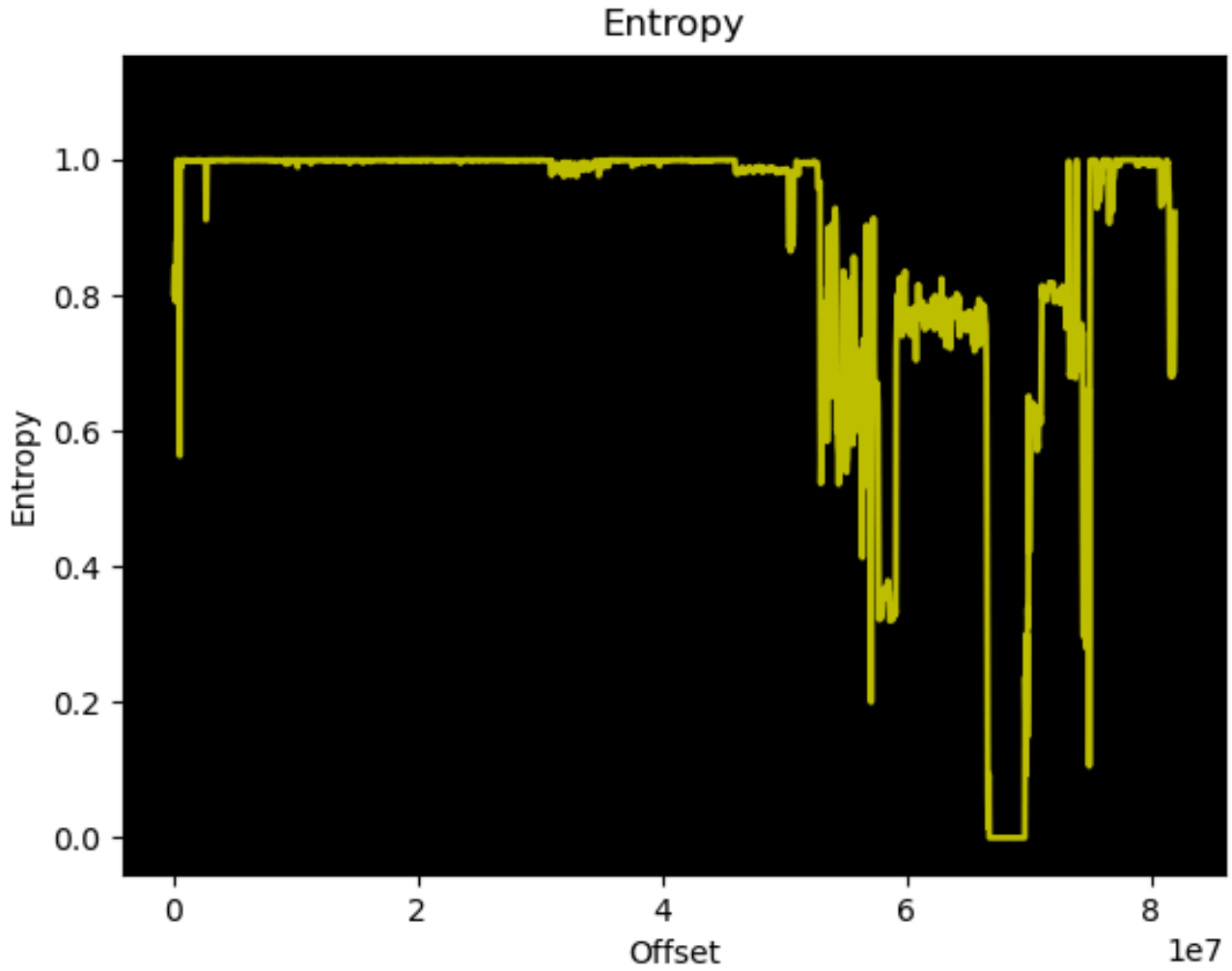
Initial Recon

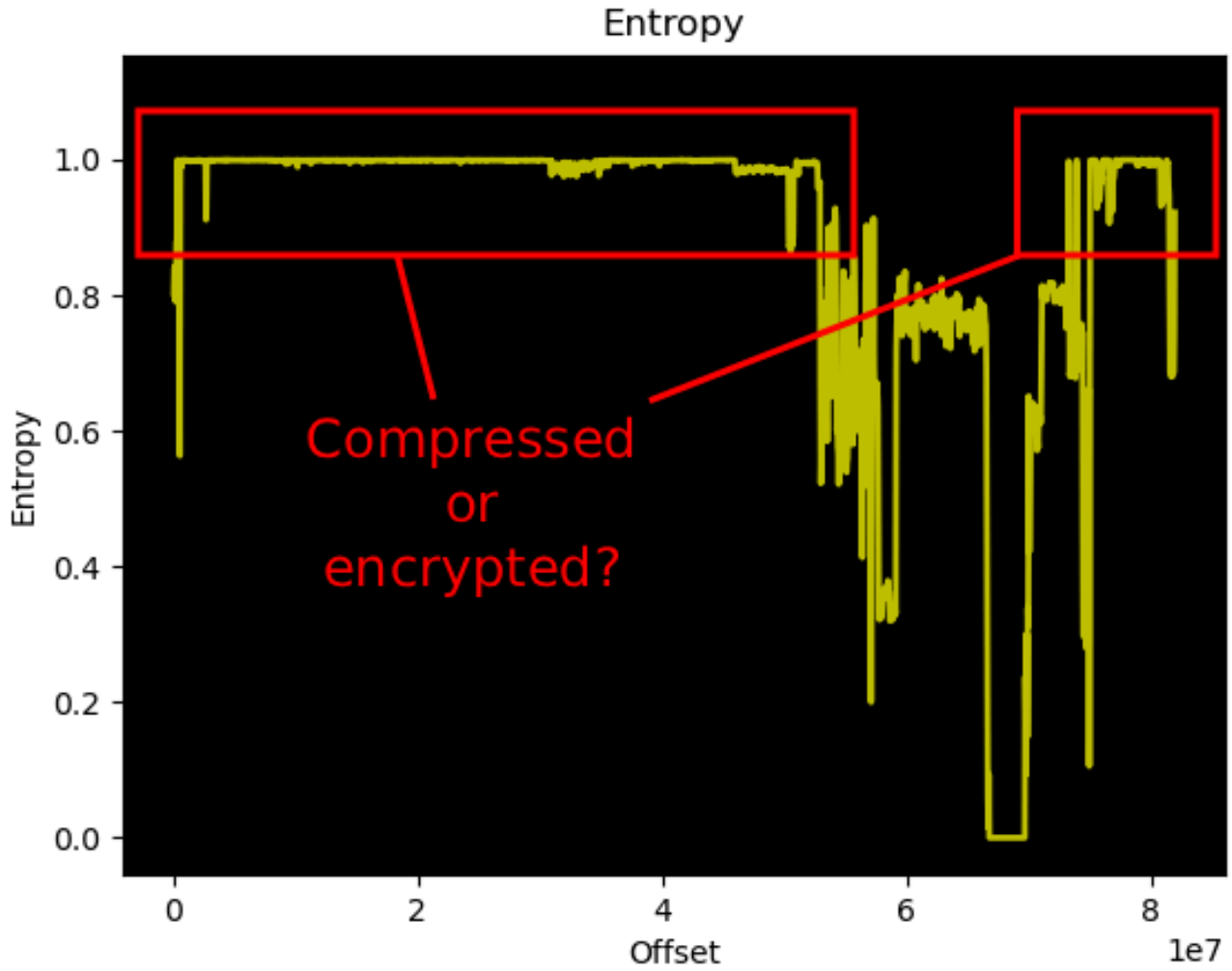
- Strings
- Entropy
- Open in Binary Ninja

```
$ ls -lh AoE2DE_s.exe | awk '{print $5}'
79M

$ strings -n10 AoE2DE_s.exe
 ...
Furor Celtica
Siege Drill
Town Watch
Crenellations
Crop Rotation
Heavy Plow
Horse Collar
 ...

$ binwalk -E AoE2DE_s.exe
```


Entropy
1.0
0.8
0.6
0.4
0.2
0.0
Entropy
0
2
4
6
8
Offset
1e7


Entropy
1.0
0.8
0.6
0.4
0.2
0.0
Entropy
Compressed
or
encrypted?
0
2
4
6
8
Offset
1e7

Sections

.text	0x140001000-0x143284400	r-x	Code
.rodata	0x143285000-0x143285800	r-x	Code
.rdata	0x143286000-0x143f86600	r--	Read-only Data
.data	0x143f87000-0x144693fe8	rw-	Writable Data
.pdata	0x144694000-0x1448a2600	r--	Read-only Data
_RDATA	0x1448a3000-0x1448a5400	r--	Read-only Data
.rsrc	0x1448a6000-0x1448c0400	r--	Read-only Data
.reloc	0x1448c1000-0x144926400	r--	Read-only Data
.idata	0x144927000-0x144933600	rw-	Writable Data
.text_2	0x144934000-0x1450fdc00	r-x	Code

Finding the decryption mechanism

- Dump decrypted binary with x64dbg
  - Reference
- Manual analysis
- Analyze decryption algorithm

```
0x142bb06c0 .text {0x140001000-0x143284400} Read-only code

142bb06c0  c0 fb a4 c0 c0 6d 8e b2-42 ee 50 7b 51 82 9d 61-6c 7c 6e dd 31 c5 cd 2e-5c 1d 54 f9 fd 90 6c 9d  .....m..B.P{Q..al|n.1...\.T...l.
142bb06e0  b6 cb d8 c6 5f cf fb 0b-00 30 74 a5 dd 98 5e 56-e9 a4 a3 aa 62 c4 a2 73-a3 e0 c2 7b 5a 40 39 ce  ...._....0t...^V....b..s...{Z@9.
142bb0700  43 00 54 27 5e a5 0c 5d-80 a1 b0 5c c8 ac f5 40-c3 2d 55 a3 5e f1 6e 50-e8 e9 4f d4 de a1 f3 6a  C.T'^..]...\...@.-U.^.nP..O....j
142bb0720  9c 38 d9 69 b1 7f 66 2c-a4 21 88 1f 26 4b 4c 2b-c2 16 27 99 a1 a5 59 5e-1d 19 9c a1 66 61 f4 9b  .8.i..f,.!..&KL+..'...Y^....fa..
142bb0740  9f f8 45 78 f1 6d 40 87-6d 04 80 8a ff 61 56 85-4f 94 d5 53 94 86 66 41-d1 a5 db 34 aa 39 58 cd  ..Ex.m@.m....aV.O..S..fA...4.9X.
142bb0760  97 a6 03 4a 87 1e 0e 11-79 ae 36 4b d4 17 6d 52-fb 86 08 c2 37 cf 42 18-ae 16 5a 6f 14 b3 63 ee  ...J....y.6K..mR....7.B...Zo..c.
142bb0780  bd 56 57 05 54 12 86 04-e0 09 c2 65 d1 06 88 2d-97 2e 64 12 28 75 a4 d1-f9 8d 0a c3 79 f5 00 7b  .VW.T......e...-..d.(u......y..{
142bb07a0  e2 82 e1 3b 97 25 39 aa-83 55 9e be 19 c6 e2 c7-97 5d 16 06 b6 5e e3 68-0f f5 27 6b c0 3b 5b db  ...;.%9..U.......]...^.h..'k.;[.
142bb07c0  90 f2 52 61 fe 6d 57 a9-90 90 48 8b                                      ..Ra.mW...H.

142bb07cc    int64_t _start()

142bb07cc  eb29              jmp     0x142bb07f7

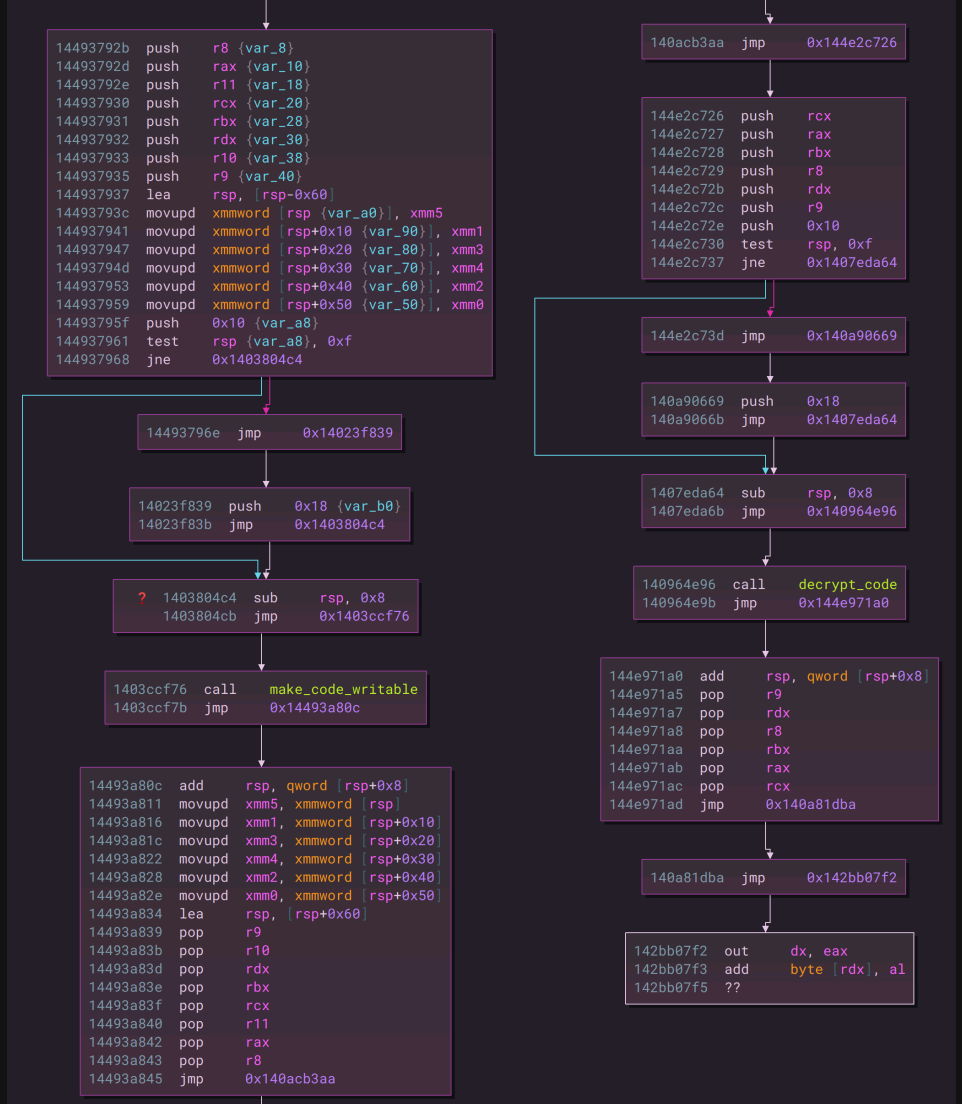
142bb07ce                                        90 8b-d0 0d 67 17 a1 6a 97 8e-ec 5e b2 3f c2 ce 90 e8              ....g..j...^.?....
142bb07e0  d5 b1 97 3e 96 58 8e 92-c8 29 aa 89 de 92 6e 64-fd db                                                  ...>.X...)....nd..

142bb07f2  ef                out     dx, eax
142bb07f3  0002              add     byte [rdx], al
142bb07f5  ce                ??

142bb07f6                                                  73                                                    s

142bb07f7  e92f71d801        jmp     0x14493792b

142bb07fc                                                                                 cc e8 df 9c                    ....
```


14493792b push r8 {var_8}
14493792d push rax {var_10}
14493792e push r11 {var_18}
144937930 push rcx {var_20}
144937931 push rbx {var_28}
144937932 push rdx {var_30}
144937933 push r10 {var_38}
144937935 push r9 {var_40}
144937937 lea rsp, [rsp-0x60]
14493793c movupd xmmword [rsp {var_a0}], xmm5
144937941 movupd xmmword [rsp+0x10 {var_90}], xmm1
144937947 movupd xmmword [rsp+0x20 {var_80}], xmm3
14493794d movupd xmmword [rsp+0x30 {var_70}], xmm4
144937953 movupd xmmword [rsp+0x40 {var_60}], xmm2
144937959 movupd xmmword [rsp+0x50 {var_50}], xmm0
14493795f push 0x10 {var_a8}
144937961 test rsp {var_a8}, 0xf
144937968 jne 0x1403804c4
14493796e jmp 0x14023f839
14023f839 push 0x18 {var_b0}
14023f83b jmp 0x1403804c4
? 1403804c4 sub rsp, 0x8
1403804cb jmp 0x1403ccf76
1403ccf76 call make_code_writable
1403ccf7b jmp 0x14493a80c
14493a80c add rsp, qword [rsp+0x8]
14493a811 movupd xmm5, xmmword [rsp]
14493a816 movupd xmm1, xmmword [rsp+0x10]
14493a81c movupd xmm3, xmmword [rsp+0x20]
14493a822 movupd xmm4, xmmword [rsp+0x30]
14493a828 movupd xmm2, xmmword [rsp+0x40]
14493a82e movupd xmm0, xmmword [rsp+0x50]
14493a834 lea rsp, [rsp+0x60]
14493a839 pop r9
14493a83b pop r10
14493a83d pop rdx
14493a83e pop rbx
14493a83f pop rcx
14493a840 pop r11
14493a842 pop rax
14493a843 pop r8
14493a845 jmp 0x140acb3aa
140acb3aa jmp 0x144e2c726
144e2c726 push rcx
144e2c727 push rax
144e2c728 push rbx
144e2c729 push r8
144e2c72b push rdx
144e2c72c push r9
144e2c72e push 0x10
144e2c730 test rsp, 0xf
144e2c737 jne 0x1407eda64
144e2c73d jmp 0x140a90669
140a90669 push 0x18
140a9066b jmp 0x1407eda64
1407eda64 sub rsp, 0x8
1407eda6b jmp 0x140964e96
140964e96 call decrypt_code
140964e9b jmp 0x144e971a0
144e971a0 add rsp, qword [rsp+0x8]
144e971a5 pop r9
144e971a7 pop rdx
144e971a8 pop r8
144e971aa pop rbx
144e971ab pop rax
144e971ac pop rcx
144e971ad jmp 0x140a81dba
140a81dba jmp 0x142bb07f2
142bb07f2 out dx, eax
142bb07f3 add byte [rdx], al
142bb07f5 ??

Exports

Search exports Aa

Ordinal	Address	Name
397	0x142309b…	AK::SoundEngine::UnregisterResourceMonitorCallback(void (__cdecl*)(struct AkResourceMonit…
398	0x1424066…	AK::SpatialAudio::UnsetGameObjectInRoom(uint64_t)
399	0x1422b79…	AK::MemoryMgr::VerifyMemoryArenaIntegrity(enum AkMemoryMgrArena)
400	0x142309c…	AK::SoundEngine::WakeupFromSuspend(uint32_t)
401	0x14234dc…	AK::WriteBytesBuffer::WriteBytes(AK::WriteBytesBuffer* this, void const* __ptr64, int32_t…
402	0x14234df…	AK::WriteBytesCount::WriteBytes(AK::WriteBytesCount* this, void const* __ptr64, int32_t, …
403	0x14234dc…	AK::WriteBytesMem::WriteBytes(AK::WriteBytesMem* this, void const* __ptr64, int32_t, int3…
404	0x144819b…	AK::SoundEngine::g_PlayingID
405	0x144819b…	g_pGameSyncMgr
406	0x144819b…	g_pRegistryMgr
407	0x14481ae…	AK::IAkStreamMgr::m_pStreamMgr
408	0x1423e65…	AkTlsAllocateSlot
409	0x1423e65…	AkTlsFreeSlot
410	0x1423e65…	AkTlsGetValue
411	0x1423e65…	AkTlsSetValue
412	0x1439de7…	AmdPowerXpressRequestHighPerformance
413	0x1439de7…	NvOptimusEnablement
414	0x14481a0…	g_pAKPluginList

Analyzing the decryption mechanism

First iteration

1. Mark code RWX
2. Decrypt with AES

- Where is the key?
- What about relocations?

Analyzing the decryption mechanism

Second iteration

1. Mark code RWX
2. Use variable-length int table to collect ciphertext
3. Validate ciphertext integrity
4. Decrypt with AES-ECB using hard-coded key
5. Put back using table

- Tail is stored separately
- Bonus: relocation-friendly stream cipher

Basic scripting to support

- Inject decrypted .text
- Copy GOT

Inject Decrypted .text

```
import lief
...
with open(TEXT_PATH, 'rb') as fin: text1_data = fin.read()
with open(TEXT2_PATH, 'rb') as fin: text2_data = fin.read()

exe = lief.parse(EXE_PATH)
text = exe.get_section('.text')

text = []
for section in exe.sections:
    if section.name == '.text':
        text.append(section)
assert len(text) == 2, len(text)

with open(EXE_PATH, 'rb') as fin:
    exe_data = bytearray(fin.read())

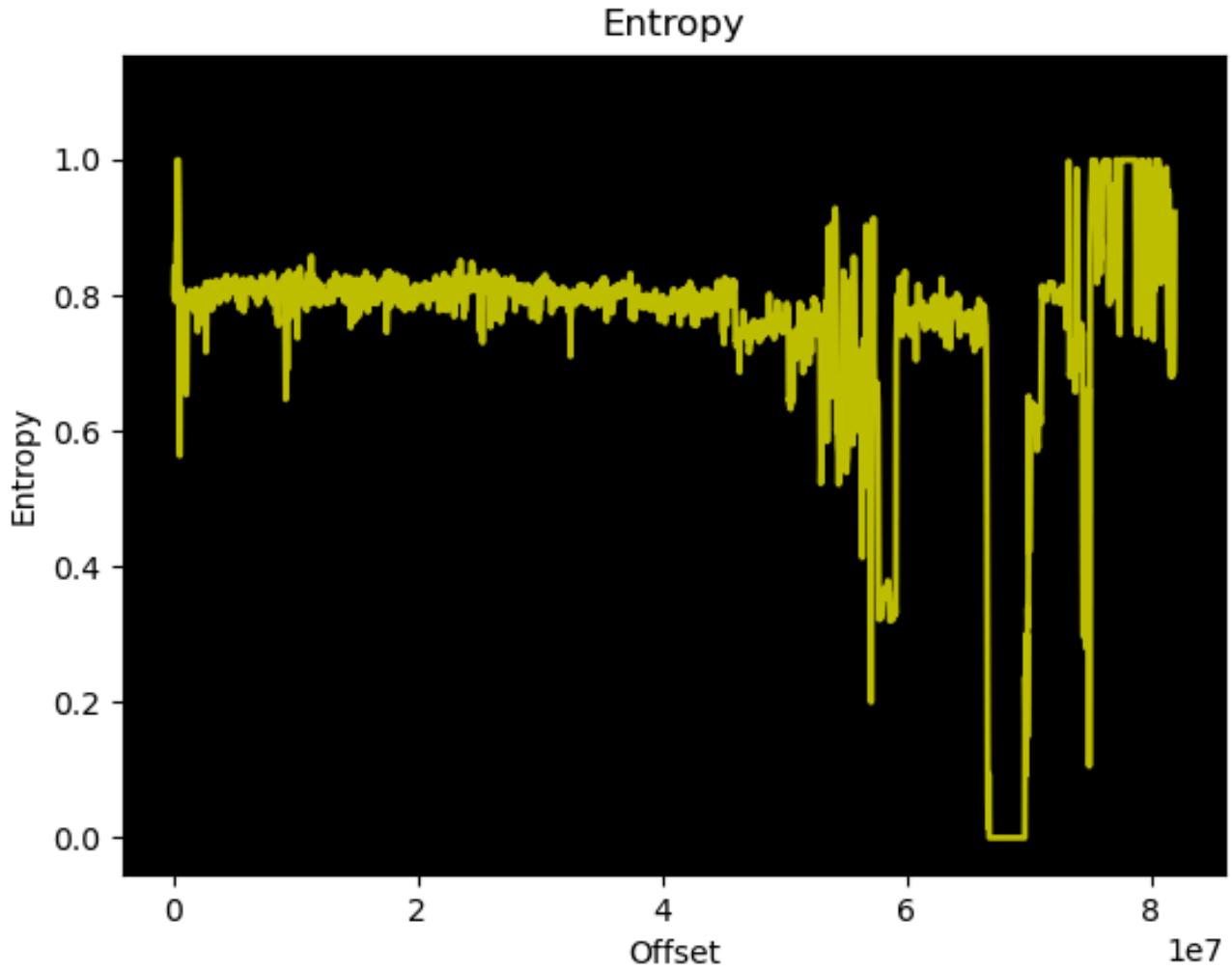
exe_data[text[0].offset:text[0].offset+text[0].size] = text1_data[:text[0].size]
exe_data[text[1].offset:text[1].offset+text[1].size] = text2_data[:text[1].size]

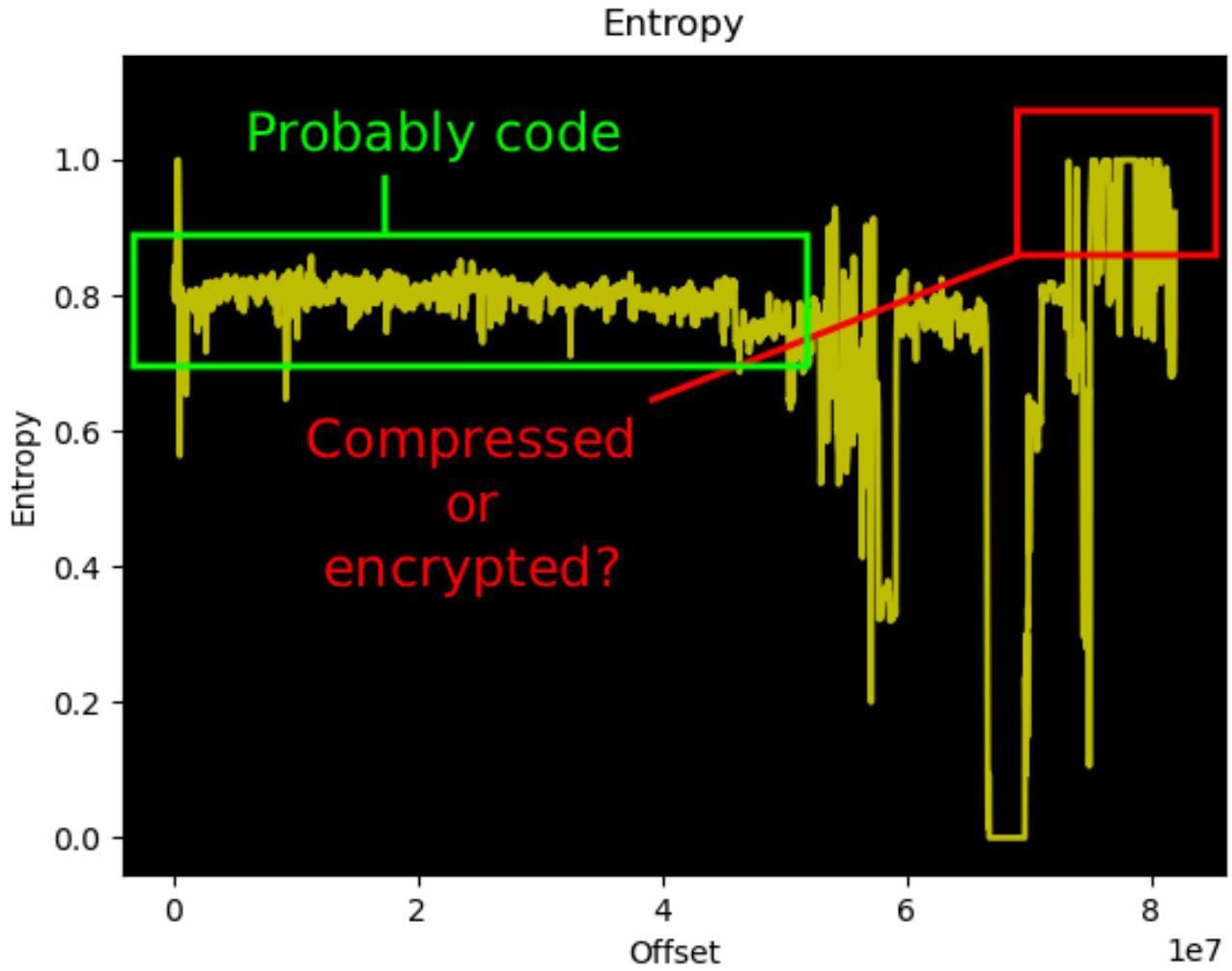
with open('AoE2DE_s.decrypted2.exe', 'wb') as fout:
    fout.write(exe_data)
```

Copy GOT in Binary Ninja

```
got_copy = [(47980544, 69856152, 41), (47980880, 69856488, 8), (47980952, 69856560, 1), ...]

BASE_ADDR = 0x140000000
for dst_addr, src_addr, num_elements in got_copy:
    for i in range(num_elements):
        src_offset = src_addr + 8*i
        dst_offset = dst_addr + 8*i
        data_var = bv.get_data_var_at(BASE_ADDR+src_offset)
        bv.define_user_data_var(BASE_ADDR+dst_offset, data_var.type, data_var.name)
        symbol = bv.get_symbol_at(BASE_ADDR+src_offset)
        bv.define_user_symbol(Symbol(symbol.type, BASE_ADDR+dst_offset, symbol.name))
        data = bv.read(BASE_ADDR+src_offset, 8)
        bv.write(BASE_ADDR+dst_offset, data)
```


Entropy
1.0
0.8
0.6
0.4
0.2
0.0
Entropy
0
2
4
6
8
Offset
1e7


Entropy
Probably code
Compressed
or
encrypted?
Entropy
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
Offset
1e7

Layer 2 Decryption

- TEA
- GOT table
- Work in progress



Feudal Age

Binary Ninja scripting





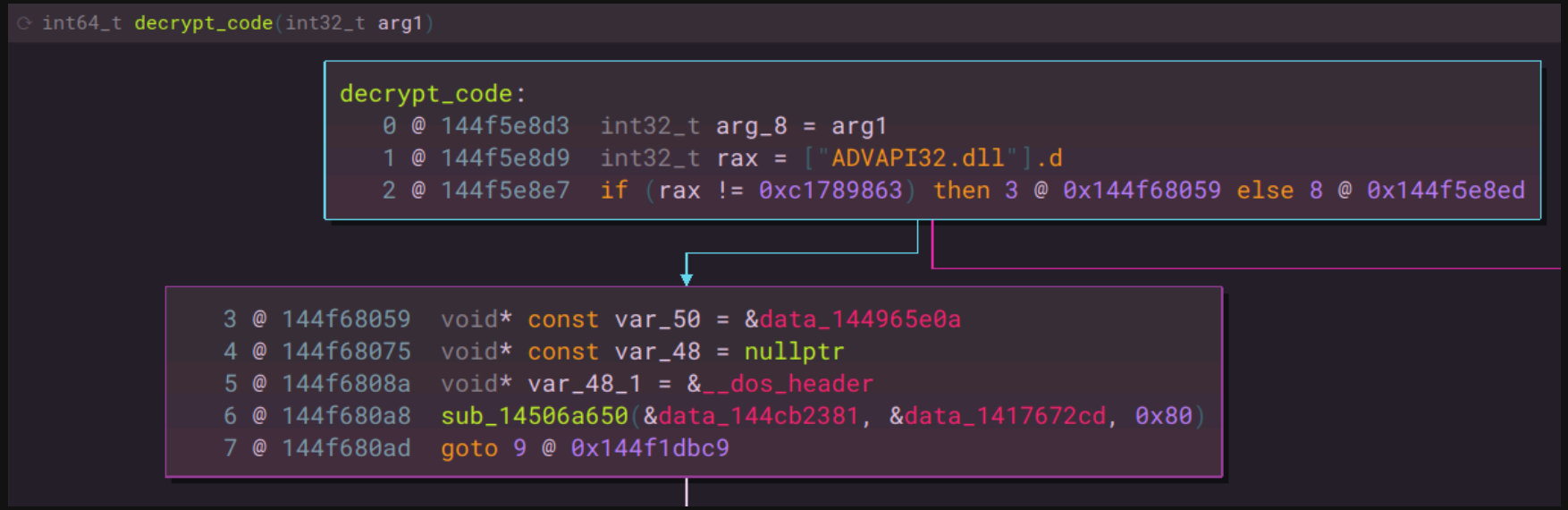
Feudal Age

Binary Ninja scripting



Automated finding of parameters

- Describe the shape of the decryption code
- Traverse IL



Automated finding of parameters

```
def aes_function_get_aes_key(aes_func: binaryninja.function.Function) → Optional[int]:
    """Anayze AES function and get AES key address"""
    for ins in aes_func.mlil.instructions:
        if ins.operation ≠ binaryninja.MediumLevelILOperation.MLIL_CALL:
            continue
        arguments = ins.operands[2]
        if len(arguments) ≠ 3:
            continue
        if arguments[2].operation ≠ binaryninja.MediumLevelILOperation.MLIL_CONST:
            continue
        if arguments[2].constant ≠ 0×80:
            continue
        if arguments[1].operation ≠ binaryninja.MediumLevelILOperation.MLIL_CONST_PTR:
            continue

        aes_key_va = arguments[1].constant
        break
    else:
        log.error("Unable to find AES setup call")
        return None
    return aes_key_va
```

Automated decryption

- Find parameters (Binary Ninja)
  - AES key VA
  - Ciphertext suffix VA
  - Ciphertext suffix len
  - Decrypt table VA
  - Target hash VA
  - Decryptor marker src VA
  - Decryptor marker dst VA
  - Sections marker VA
- Decrypt (lief)

Find Offsets

```
$ python3 find-offsets.py AoE2DE_s.exe
Processing AoE2 exe at "AoE2DE_s.exe"
Entry function: 0×142bb07cc
Sections mod function: 0×140a8dcbc
Code decrypt function: 0×144f5e8b8
Section marker VA: 0×140526f5a
Decryption marker source VA: 0×140cbbcaa
Decryption marker destination VA: 0×1449297ae
AES key VA: 0×1417672cd
Target hash VA: 0×144f3cab3
Decryption table VA: 0×144965e0a
Ciphertext suffix VA: 0×144ea9e1f
Ciphertext suffix Length: 0×a
```

Decrypt Binary

```
$ python3 decrypt-binary.py --input-path AoE2DE_s.exe --output-path AoE2DE_s.decrypted.exe --binary-parameters '{
    "aes_key_va": 5393248973,
    "ciphertext_suffix_va": 5451193887,
    "ciphertext_suffix_len": 10,
    "decrypt_table_va": 5445672458,
    "decryptor_marker_src_va": 5382061226,
    "decryptor_marker_dst_va": 5445425070,
    "target_hash_va": 5451795123,
    "sections_marker_va": 5374111578,
    "decrypt_table_data_src_va": null,
    "decrypt_table_data_dst_va": null,
    "got_marker_va": null
}'
Decryption table, VA: 0×144965e0a, offset: 0×469d80a
Ciphertext hash ok: 1d9fd9a570a1c7d20bb35880d59113fe2b9ed33b8f46282b4685e34614340a6f2bbebeee7ab353a3b2b50b2745e32336
Ciphertext suffix: 6c704cf2baf01917617e
Decrypted: 4053488b442440488bd94885c0745a4c ...
```

Leveraging Steam DepotDownloader for CI

- Download every version of the game
- BONUS: unencrypted version
- Find decryption parameter for every version
- Decrypt every version of the game

Mass Find Decryption Parameters

```
$ python3 ./scripts/find_offsets_mass.py
INFO processing 8706369178041085088
INFO processing 6991182055832393170
ERROR Inconsistent decryption marker situation
ERROR failed to find parameters for version 6991182055832393170
INFO processing 3568369477204818532
INFO processing 8916508292174982468
INFO processing 620891448408726573
INFO processing 7959981682396439424
INFO processing 2176586667924974355
INFO processing 7673739171887156079
INFO processing 2164364554796430252
ERROR Unable to find suffix function
ERROR failed to find parameters for version 2164364554796430252
...
INFO successes: 80, failures: 16
```

Mass Decrypt Binaries

```
$ python3 ./scripts/decrypt_binary_mass.py
INFO processing 8706369178041085088
INFO Decryption table, VA: 0×144363b67, offset: 0×3e5bd67
INFO ciphertext hash valid: 0d8d7515c2cf711386b71c9191046339c8d55776825415296ad6ca93629ecf8c ...
INFO Ciphertext suffix: 5a9bebbce3f79383a7610a
INFO Decrypted: 48897c2408488974241048894e04889cf ...
INFO processing 6991182055832393170
WARNING version 6991182055832393170 parameters do not exist
INFO processing 3568369477204818532
INFO Decryption table, VA: 0×144446e5c, offset: 0×420b45c
INFO ciphertext hash valid: 98ee3c8e5f8454985ed1c2dde1e79cc64442aa5cfa1df0a14d853406ddcbadb9 ...
INFO Ciphertext suffix: a4415ba8f51cffbfde4b
INFO Decrypted: 4053488b442440488bd94885c0745a4c ...
INFO processing 8425935300679735926
INFO Decryption table, VA: 0×143953156, offset: 0×353b356
ERROR ciphertext hash invalid: b83bf9ecc45469eac1b4da02e811ba71cf6decf84d35c63cc082fff5ee166105 ...
ERROR failed to decrypt version 8425935300679735926
...
INFO successes: 76, failures: 20
```



Castle Age

BNIL query system



Castle Age

BNIL query system

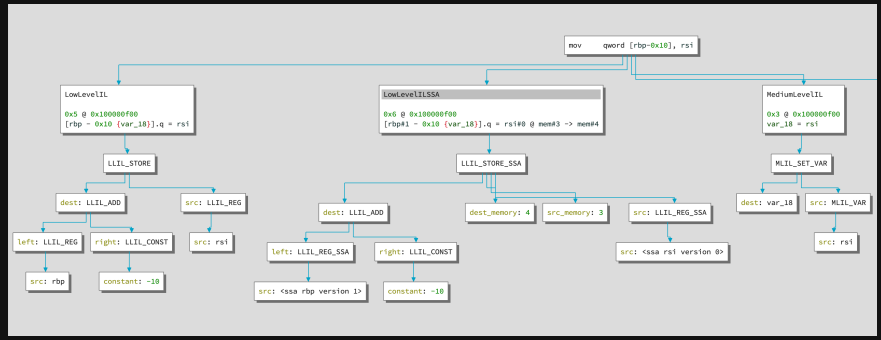


Pattern match obfuscations with tree sitter

- Inspiration
  - BNIL Instruction Graph by Ryan Stortz
  - Weggli by Felix Wilhelm
- Create lisp-like syntax
- Parse with tree sitter

```
weggli '{
    _ $buf[_];
    memcpy($buf,_,_);
}' ./target/src

weggli '{
    $ret = snprintf($b,_,_);
    $b[$ret] = _;
}' ./target/src
```



Example: Obfuscated Code

Original:

```
push 0×1337
```

Obfuscated:

```
sub rsp, 8
mov [rsp], rcx
mov rcx, 0×1337
xchg rcx, [rsp]
```

LLIL:

```
rsp = rsp - 8
*rsp = rcx
rcx = 0×1337
tmp = rcx
rcx = *rsp
*rsp = tmp
```

Example: BNIL Tree Sitter Query

```
(
    (llil_instruction (constant) @idx_1
        (llil_store dest: (llil_reg (identifier) @_rsp1) src: (llil_reg src: (identifier) @_reg_a1))) .
    (llil_instruction (constant) @idx_2
        (llil_set_reg dest: (identifier) @_reg_a2 src: (_ constant: (constant) @constant))) .
    (llil_instruction (constant) @idx_3
        (llil_set_reg dest: (identifier) @_reg_b1 src: (llil_load (llil_reg (identifier) @_rsp1)))) .
    (llil_instruction (constant) @idx_4
        (llil_store dest: (llil_reg (identifier) @_rsp2) src: (llil_reg src: (identifier) @_reg_a3))) .
    (llil_instruction (constant) @idx_5
        (llil_set_reg dest: (identifier) @_reg_a4 src: (llil_reg src: (identifier) @_reg_b2))) .

    (#eq? @_reg_a1 @_reg_a2)
    (#eq? @_reg_a2 @_reg_a3)
    (#eq? @_reg_a3 @_reg_a4)
    (#eq? @_reg_b1 @_reg_b2)
    (#eq? @_rsp1 "rsp")
    (#eq? @_rsp2 "rsp")
)
```

BNIL modification to deobfuscate

- Query for patterns
- NOP matched instructions
- Replace last instruction
- Loop until no matches

Example: LLIL identification code

```
jmp_srcs, jmp_dests = {}, {}
for pattern in JUMP_CONST_PATTERNS:
    matches = bnil_matcher.bnil.match_llil(context.function, pattern)
    for _, nodes in matches:
        made_change, match_idxs = True, set()

        for label, node in nodes.items():
            if label.startswith("_"):
                continue
            if label.startswith("idx_"):
                match_idxs.add(int(node[0].text.decode(), 16))

        jmp_dest = int(nodes["constant"][0].text.decode(), 16)
        jmp_dest_instructions = old_func.get_instructions_at(jmp_dest)
        jmp_src_idx = max(match_idxs)
        if jmp_dest_instructions:
            jmp_dest_idx = jmp_dest_instructions[0]
            if not (label := old_func.get_label_for_source_instruction(jmp_dest_idx)):
                label = LowLevelILLabel()
                jmp_dests[jmp_dest_idx] = label
            jmp_srcs[jmp_src_idx] = (True, label)
        else:
            jmp_srcs[jmp_src_idx] = (False, jmp_dest)
```

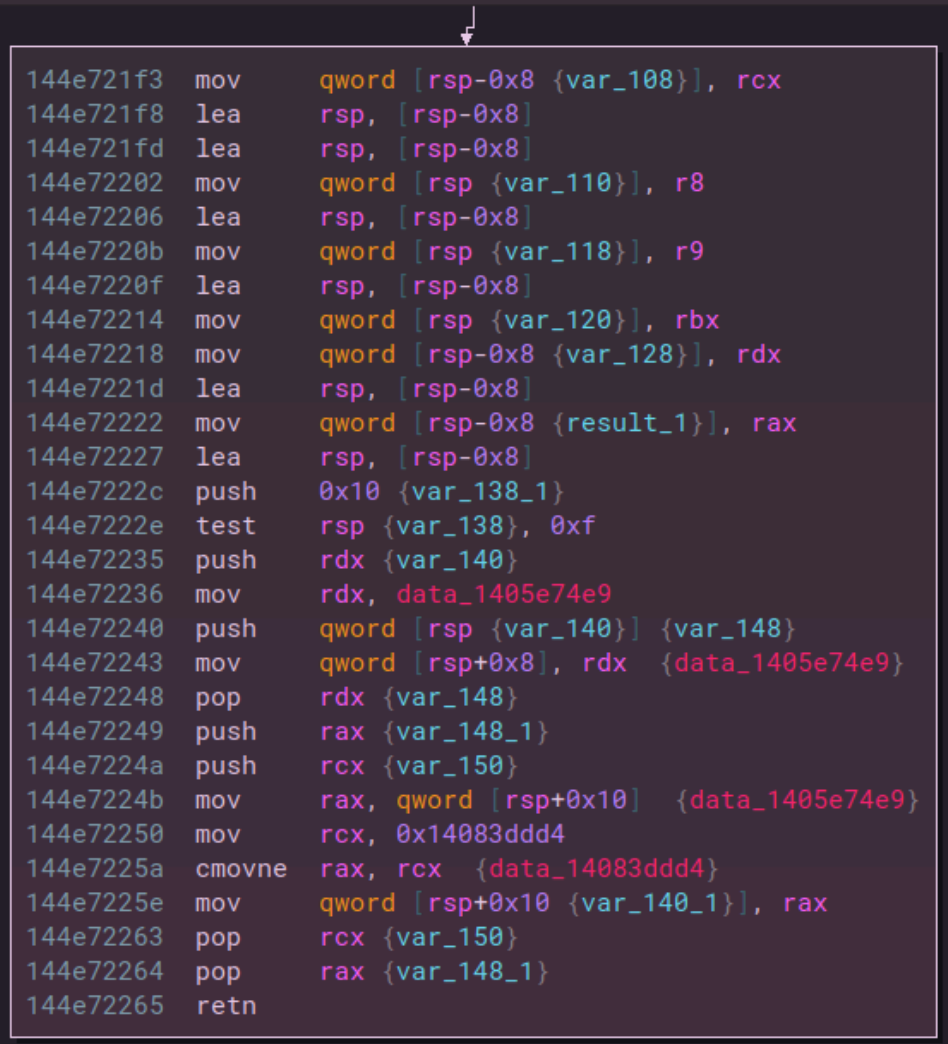
Example: LLIL modification code

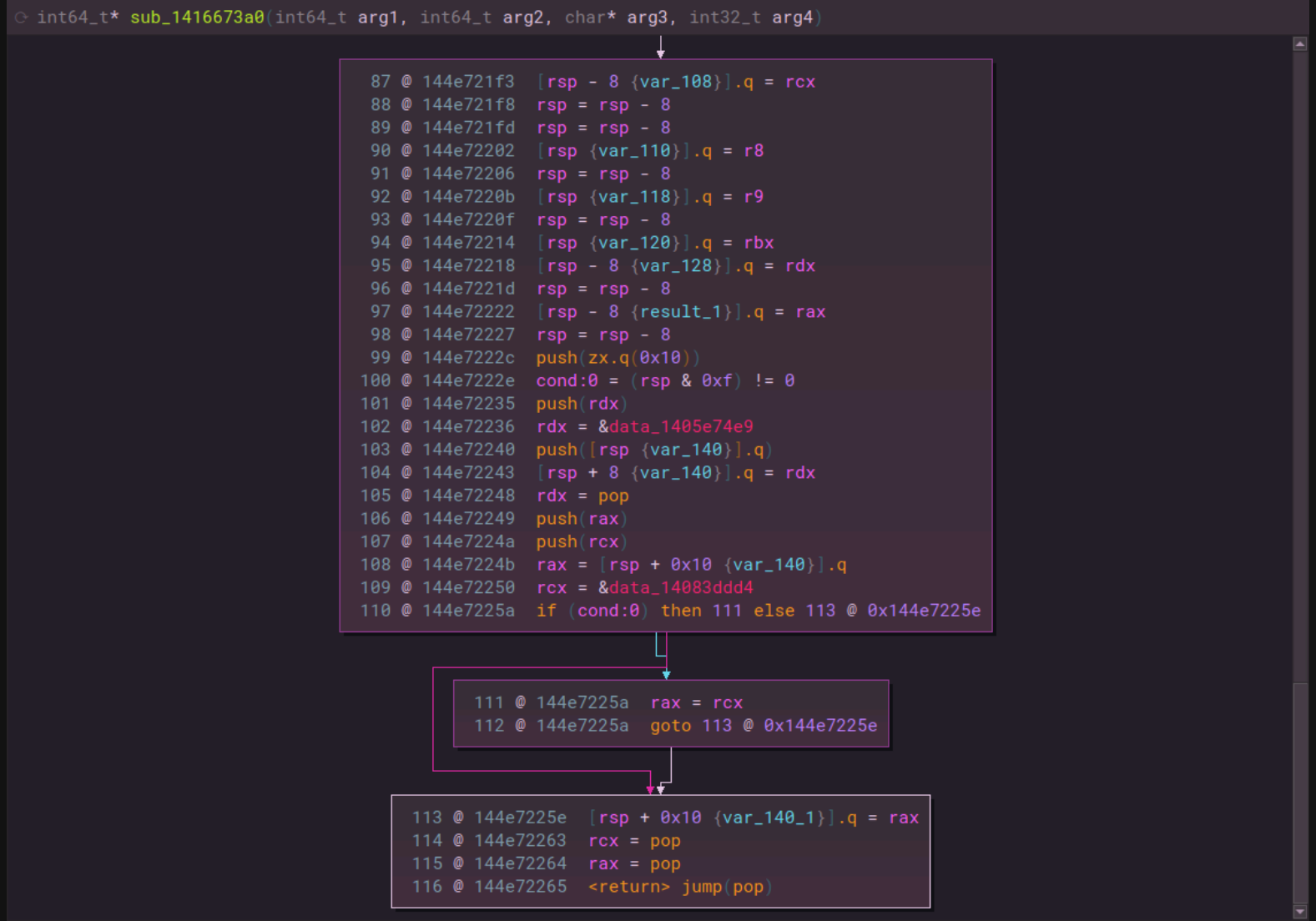
```
for old_instr_index in range(old_block.start, old_block.end):
    old_instr: LowLevelILInstruction = old_func[old_instr_index]
    new_func.set_current_address(old_instr.address, old_block.arch)

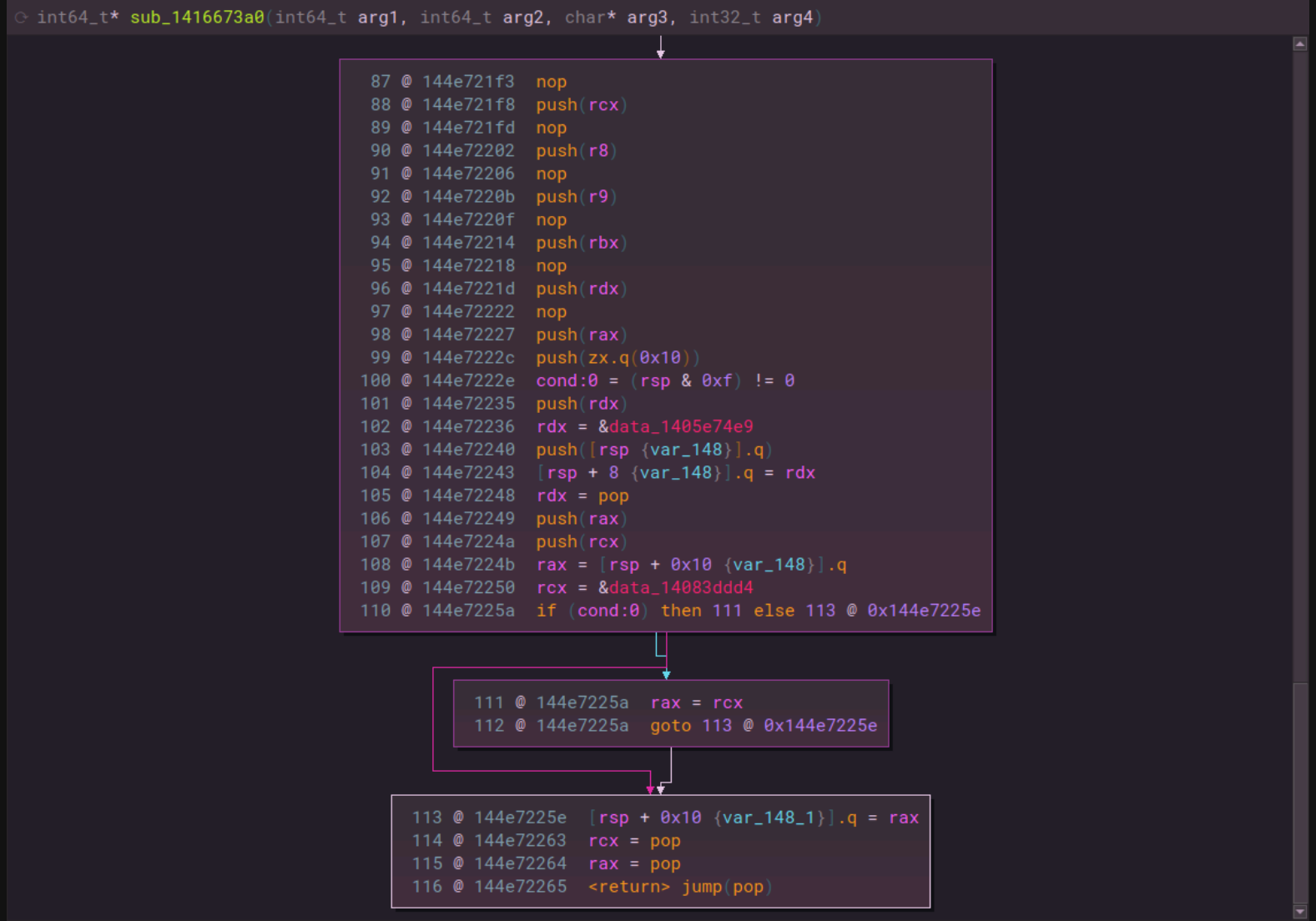
    if label := jmp_dests.get(old_instr_index, None):
        print(f"DST LABEL {label}")
        new_func.mark_label(label)

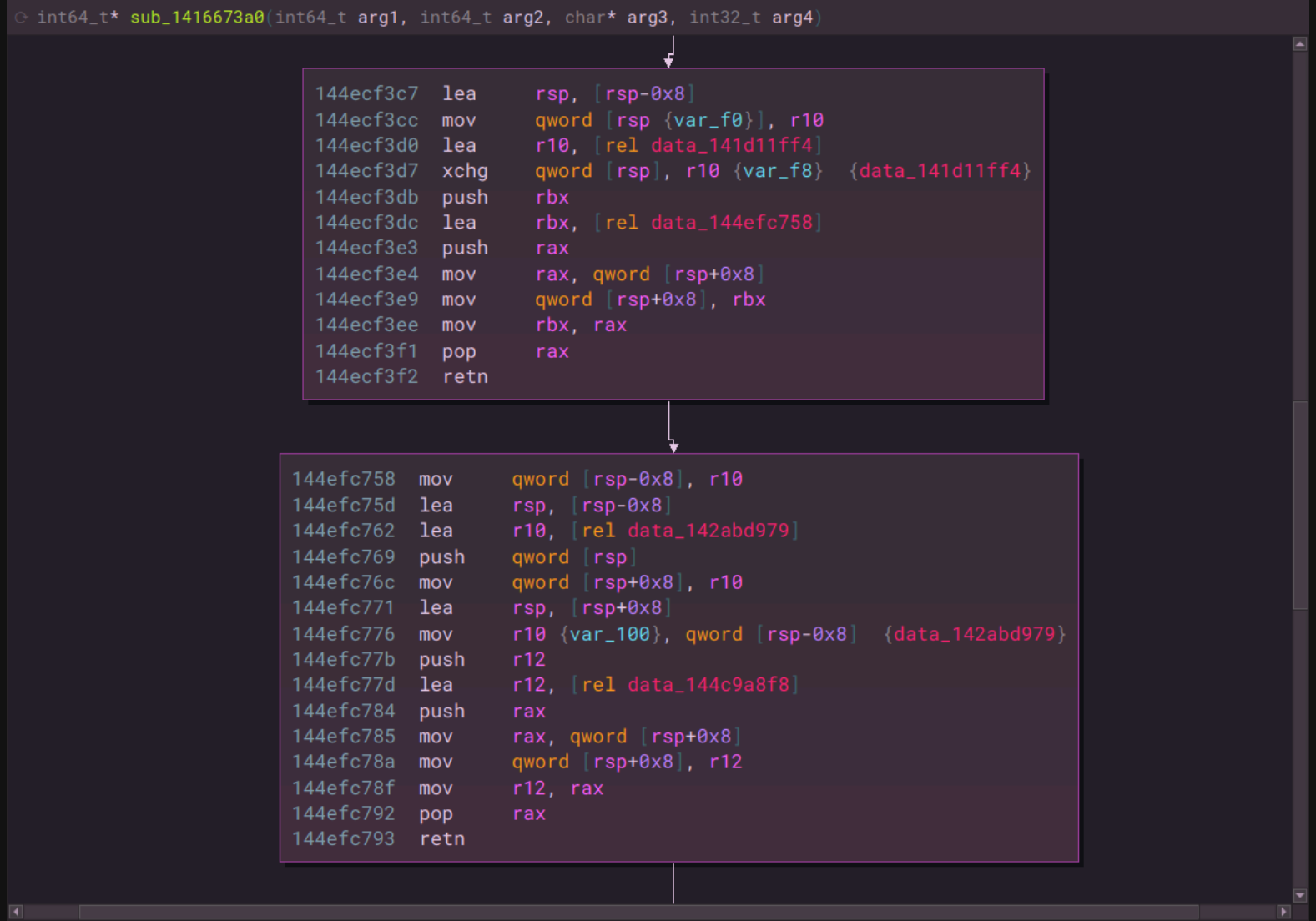
    if jmp_src := jmp_srcs.get(old_instr_index, None):
        in_func, jmp_data = jmp_src
        if in_func:
            print(f"GOTO LABEL {jmp_data}")
            new_expr = new_func.goto(jmp_data, ILSourceLocation.from_instruction(old_instr))
            new_func.append(new_expr)
            continue
        else:
            new_expr = new_func.jump(
                new_func.const_pointer(8, jmp_data, ILSourceLocation.from_instruction(old_instr)),
                ILSourceLocation.from_instruction(old_instr),
            )
            new_func.append(new_expr)
            continue
```

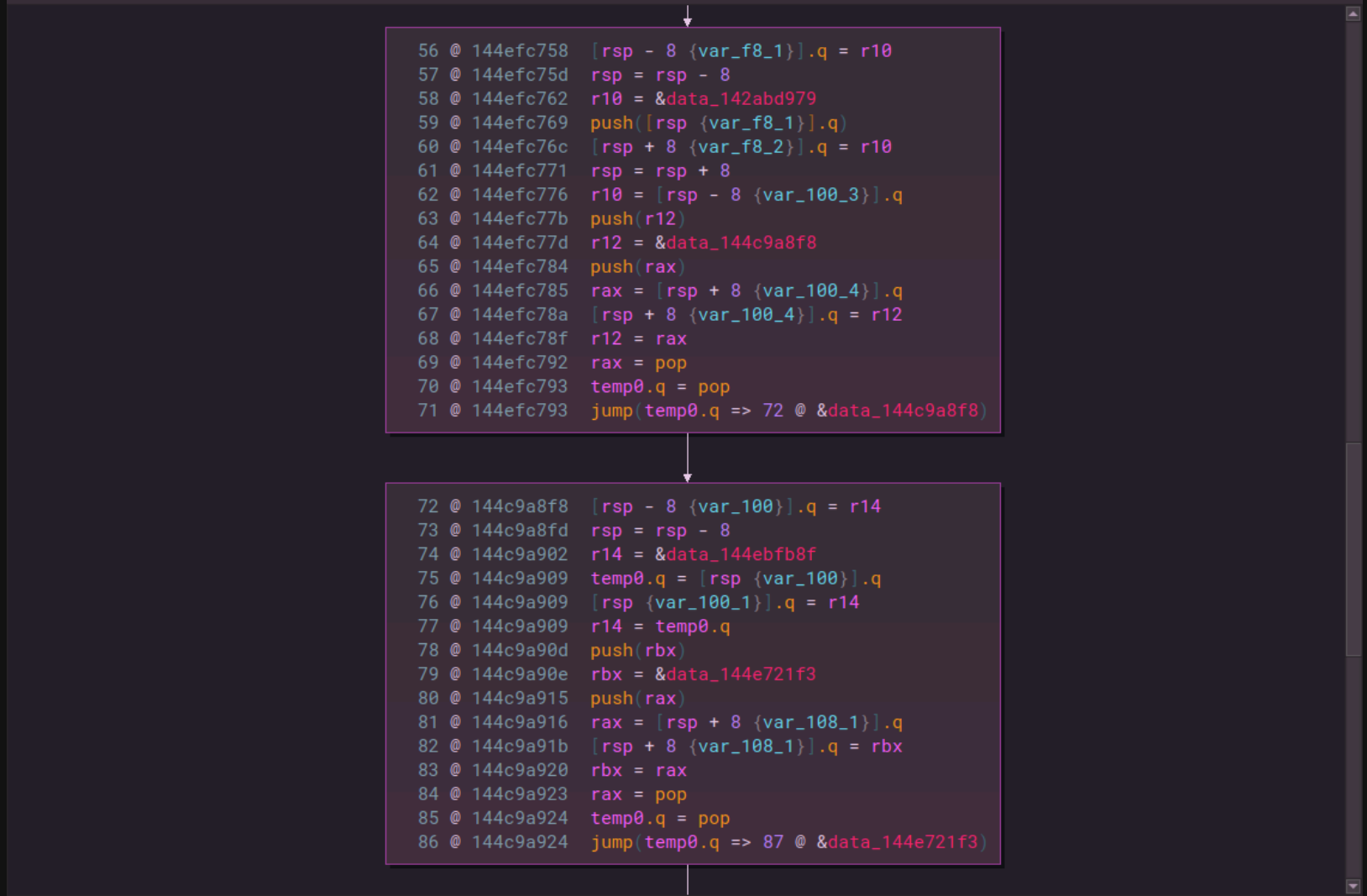
```
int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
```




int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
87 @ 144e721f3 [rsp - 8 {var_108}].q = rcx
88 @ 144e721f8 rsp = rsp - 8
89 @ 144e721fd rsp = rsp - 8
90 @ 144e72202 [rsp {var_110}].q = r8
91 @ 144e72206 rsp = rsp - 8
92 @ 144e7220b [rsp {var_118}].q = r9
93 @ 144e7220f rsp = rsp - 8
94 @ 144e72214 [rsp {var_120}].q = rbx
95 @ 144e72218 [rsp - 8 {var_128}].q = rdx
96 @ 144e7221d rsp = rsp - 8
97 @ 144e72222 [rsp - 8 {result_1}].q = rax
98 @ 144e72227 rsp = rsp - 8
99 @ 144e7222c push(zx.q(0x10))
100 @ 144e7222e cond:0 = (rsp & 0xf) != 0
101 @ 144e72235 push(rdx)
102 @ 144e72236 rdx = &data_1405e74e9
103 @ 144e72240 push([rsp {var_140}].q)
104 @ 144e72243 [rsp + 8 {var_140}].q = rdx
105 @ 144e72248 rdx = pop
106 @ 144e72249 push(rax)
107 @ 144e7224a push(rcx)
108 @ 144e7224b rax = [rsp + 0x10 {var_140}].q
109 @ 144e72250 rcx = &data_14083ddd4
110 @ 144e7225a if (cond:0) then 111 else 113 @ 0x144e7225e
111 @ 144e7225a rax = rcx
112 @ 144e7225a goto 113 @ 0x144e7225e
113 @ 144e7225e [rsp + 0x10 {var_140_1}].q = rax
114 @ 144e72263 rcx = pop
115 @ 144e72264 rax = pop
116 @ 144e72265 <return> jump(pop)

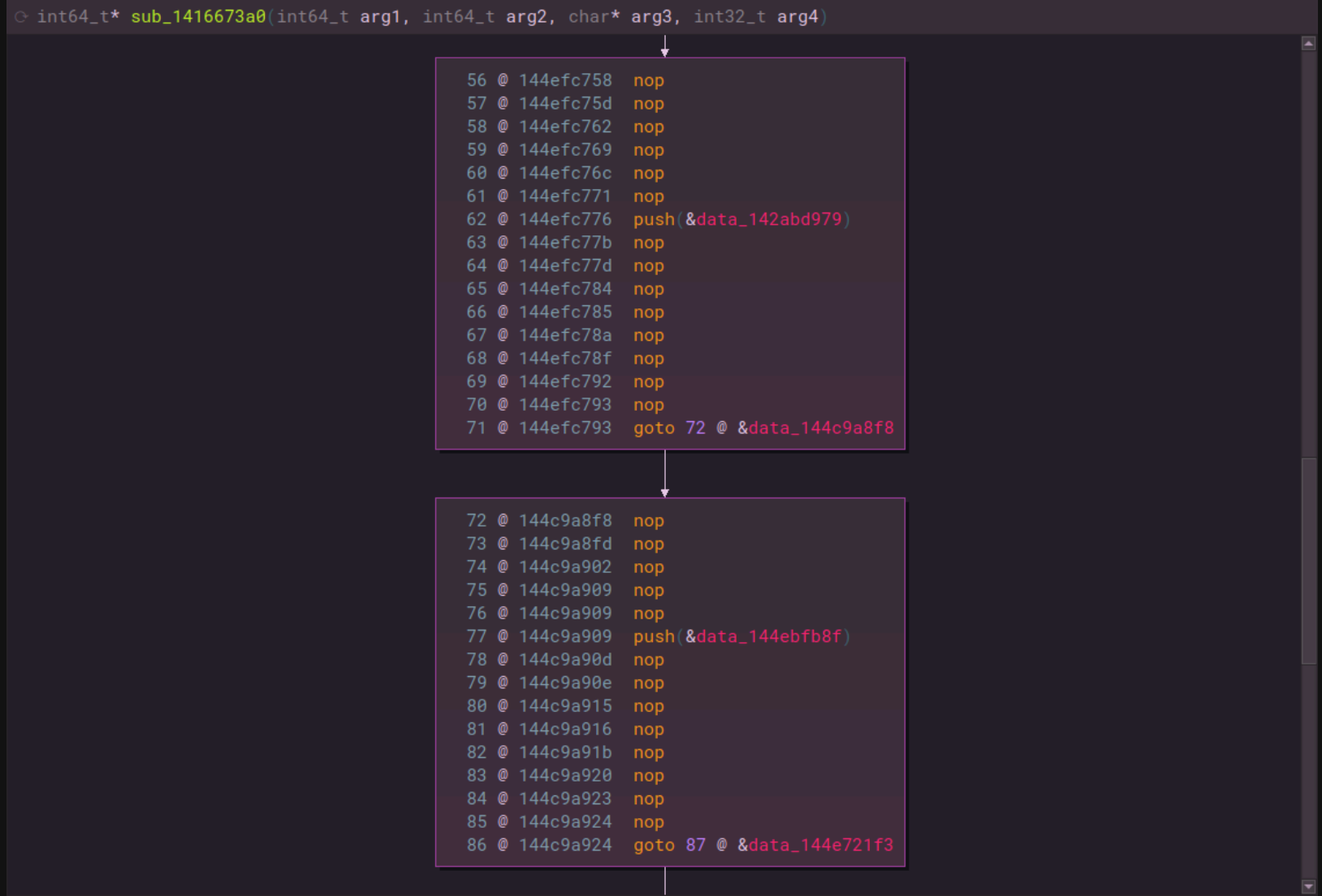

int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
87 @ 144e721f3 nop
88 @ 144e721f8 push(rcx)
89 @ 144e721fd nop
90 @ 144e72202 push(r8)
91 @ 144e72206 nop
92 @ 144e7220b push(r9)
93 @ 144e7220f nop
94 @ 144e72214 push(rbx)
95 @ 144e72218 nop
96 @ 144e7221d push(rdx)
97 @ 144e72222 nop
98 @ 144e72227 push(rax)
99 @ 144e7222c push(zx.q(0x10))
100 @ 144e7222e cond:0 = (rsp & 0xf) != 0
101 @ 144e72235 push(rdx)
102 @ 144e72236 rdx = &data_1405e74e9
103 @ 144e72240 push([rsp {var_148}].q)
104 @ 144e72243 [rsp + 8 {var_148}].q = rdx
105 @ 144e72248 rdx = pop
106 @ 144e72249 push(rax)
107 @ 144e7224a push(rcx)
108 @ 144e7224b rax = [rsp + 0x10 {var_148}].q
109 @ 144e72250 rcx = &data_14083ddd4
110 @ 144e7225a if (cond:0) then 111 else 113 @ 0x144e7225e
111 @ 144e7225a rax = rcx
112 @ 144e7225a goto 113 @ 0x144e7225e
113 @ 144e7225e [rsp + 0x10 {var_148_1}].q = rax
114 @ 144e72263 rcx = pop
115 @ 144e72264 rax = pop
116 @ 144e72265 <return> jump(pop)

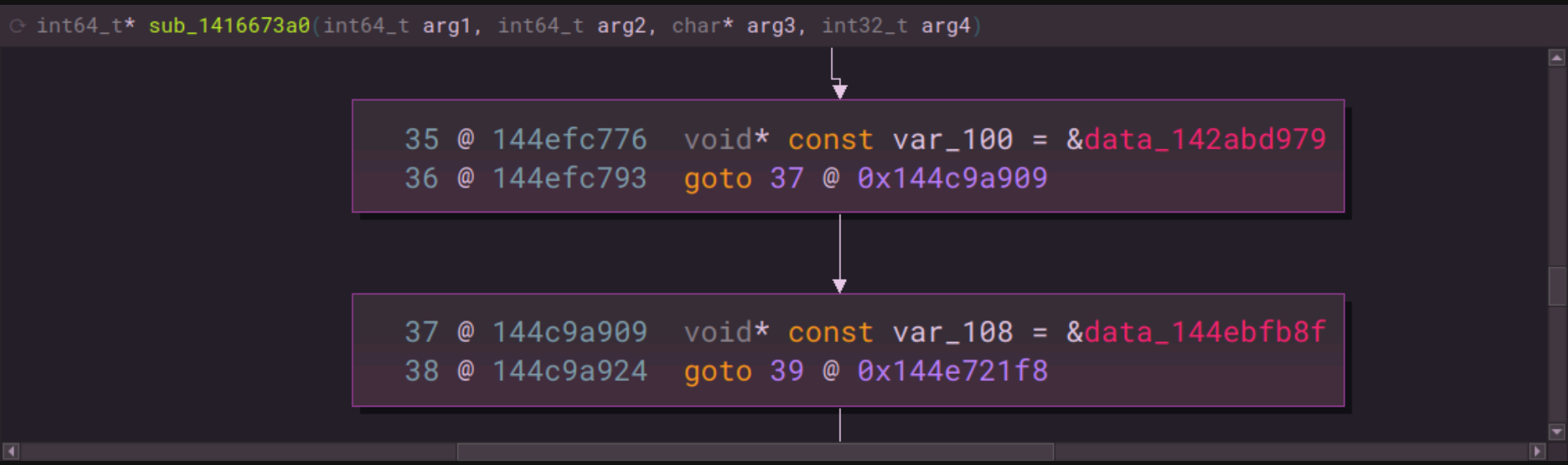

int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
144ecf3c7 lea rsp, [rsp-0x8]
144ecf3cc mov qword [rsp {var_f0}], r10
144ecf3d0 lea r10, [rel data_141d11ff4]
144ecf3d7 xchg qword [rsp], r10 {var_f8} {data_141d11ff4}
144ecf3db push rbx
144ecf3dc lea rbx, [rel data_144efc758]
144ecf3e3 push rax
144ecf3e4 mov rax, qword [rsp+0x8]
144ecf3e9 mov qword [rsp+0x8], rbx
144ecf3ee mov rbx, rax
144ecf3f1 pop rax
144ecf3f2 retn
144efc758 mov qword [rsp-0x8], r10
144efc75d lea rsp, [rsp-0x8]
144efc762 lea r10, [rel data_142abd979]
144efc769 push qword [rsp]
144efc76c mov qword [rsp+0x8], r10
144efc771 lea rsp, [rsp+0x8]
144efc776 mov r10 {var_100}, qword [rsp-0x8] {data_142abd979}
144efc77b push r12
144efc77d lea r12, [rel data_144c9a8f8]
144efc784 push rax
144efc785 mov rax, qword [rsp+0x8]
144efc78a mov qword [rsp+0x8], r12
144efc78f mov r12, rax
144efc792 pop rax
144efc793 retn


int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
56 @ 144efc758 [rsp - 8 {var_f8_1}].q = r10
57 @ 144efc75d rsp = rsp - 8
58 @ 144efc762 r10 = &data_142abd979
59 @ 144efc769 push([rsp {var_f8_1}].q)
60 @ 144efc76c [rsp + 8 {var_f8_2}].q = r10
61 @ 144efc771 rsp = rsp + 8
62 @ 144efc776 r10 = [rsp - 8 {var_100_3}].q
63 @ 144efc77b push(r12)
64 @ 144efc77d r12 = &data_144c9a8f8
65 @ 144efc784 push(rax)
66 @ 144efc785 rax = [rsp + 8 {var_100_4}].q
67 @ 144efc78a [rsp + 8 {var_100_4}].q = r12
68 @ 144efc78f r12 = rax
69 @ 144efc792 rax = pop
70 @ 144efc793 temp0.q = pop
71 @ 144efc793 jump(temp0.q => 72 @ &data_144c9a8f8)
72 @ 144c9a8f8 [rsp - 8 {var_100}].q = r14
73 @ 144c9a8fd rsp = rsp - 8
74 @ 144c9a902 r14 = &data_144ebfb8f
75 @ 144c9a909 temp0.q = [rsp {var_100}].q
76 @ 144c9a909 [rsp {var_100_1}].q = r14
77 @ 144c9a909 r14 = temp0.q
78 @ 144c9a90d push(rbx)
79 @ 144c9a90e rbx = &data_144e721f3
80 @ 144c9a915 push(rax)
81 @ 144c9a916 rax = [rsp + 8 {var_108_1}].q
82 @ 144c9a91b [rsp + 8 {var_108_1}].q = rbx
83 @ 144c9a920 rbx = rax
84 @ 144c9a923 rax = pop
85 @ 144c9a924 temp0.q = pop
86 @ 144c9a924 jump(temp0.q => 87 @ &data_144e721f3)

Serialized LLIL

```
0×24: (LLIL_SET_REG 0×76 rsp (LLIL_ADD 0×75 (LLIL_REG 0×74 rsp) (LLIL_CONST 0×73 -0×8)))
0×25: (LLIL_STORE 0×79 (LLIL_REG 0×78 rsp) (LLIL_REG 0×77 r10))
0×26: (LLIL_SET_REG 0×7b r10 (LLIL_CONST_PTR 0×7a 0×141d11ff4))
0×27: (LLIL_SET_REG 0×7e temp0 (LLIL_LOAD 0×7d (LLIL_REG 0×7c rsp)))
0×28: (LLIL_STORE 0×81 (LLIL_REG 0×80 rsp) (LLIL_REG 0×7f r10))
0×29: (LLIL_SET_REG 0×83 r10 (LLIL_REG 0×82 temp0))
0×2a: (LLIL_PUSH 0×85 (LLIL_REG 0×84 rbx))
0×2b: (LLIL_SET_REG 0×87 rbx (LLIL_CONST_PTR 0×86 0×144efc758))
0×2c: (LLIL_PUSH 0×89 (LLIL_REG 0×88 rax))
0×2d: (LLIL_SET_REG 0×8e rax (LLIL_LOAD 0×8d (LLIL_ADD 0×8c (LLIL_REG 0×8b rsp) (LLIL_CONST 0×8a 0×8))))
0×2e: (LLIL_STORE 0×93 (LLIL_ADD 0×92 (LLIL_REG 0×91 rsp) (LLIL_CONST 0×90 0×8)) (LLIL_REG 0×8f rbx))
0×2f: (LLIL_SET_REG 0×95 rbx (LLIL_REG 0×94 rax))
0×30: (LLIL_SET_REG 0×97 rax (LLIL_POP 0×96 ))
0×31: (LLIL_SET_REG 0×99 temp0 (LLIL_POP 0×98 ))
0×32: (LLIL_JUMP_TO 0×9c (LLIL_REG 0×9a temp0) (map (5451532120 56)))
```


int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
56 @ 144efc758 nop
57 @ 144efc75d nop
58 @ 144efc762 nop
59 @ 144efc769 nop
60 @ 144efc76c nop
61 @ 144efc771 nop
62 @ 144efc776 push(&data_142abd979)
63 @ 144efc77b nop
64 @ 144efc77d nop
65 @ 144efc784 nop
66 @ 144efc785 nop
67 @ 144efc78a nop
68 @ 144efc78f nop
69 @ 144efc792 nop
70 @ 144efc793 nop
71 @ 144efc793 goto 72 @ &data_144c9a8f8
72 @ 144c9a8f8 nop
73 @ 144c9a8fd nop
74 @ 144c9a902 nop
75 @ 144c9a909 nop
76 @ 144c9a909 nop
77 @ 144c9a909 push(&data_144ebfb8f)
78 @ 144c9a90d nop
79 @ 144c9a90e nop
80 @ 144c9a915 nop
81 @ 144c9a916 nop
82 @ 144c9a91b nop
83 @ 144c9a920 nop
84 @ 144c9a923 nop
85 @ 144c9a924 nop
86 @ 144c9a924 goto 87 @ &data_144e721f3


int64_t* sub_1416673a0(int64_t arg1, int64_t arg2, char* arg3, int32_t arg4)
35 @ 144efc776 void* const var_100 = &data_142abd979
36 @ 144efc793 goto 37 @ 0x144c9a909
37 @ 144c9a909 void* const var_108 = &data_144ebfb8f
38 @ 144c9a924 goto 39 @ 0x144e721f8

Instrumentation to verify

- Sogen
  - Windows user space emulator
  - Maurice Heumann
  - Talk at Navaja Negra 2025
- Validate decryption
  - Currently: some versions ok, some crash
- Investigate anti-tampering
- Develop tools

Imperial Age

What's next?



Imperial Age

What's next?



Future ideas

- Headless game engine
  - Game replay analysis
  - AI development
- Vulnerability research
  - Fuzzing?
  - RedRocket blog post, 2021

Conclusion

- Started with encrypted binary
- Decryption
- Automation pipeline
- BNIL query system

Thank You!