

```

loc_182489c0
PUSH    RBP
PUSH    R15
PUSH    R14
PUSH    R13
PUSH    R12
PUSH    RSI
PUSH    RDI
PUSH    RBX
SUB      RSP, 0x68
LEA      RBP, QWORD PTR [RSP + 0x60]
MOV      QWORD PTR [RBP], 0xFFFFFFFFFFFFFFF0
LEA      RCX, QWORD PTR [RIP + 0x113C29F]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0], RCX
ADD      ECX, 0x89F4
ADD      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0], RCX
MOV      RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
MOV      R8D, DWORD PTR [RCX + 0xFFFFFFFFF3A49]
JLEA     QWORD PTR [RIP + 0xDC9C9C]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], RCX
MOV      RCX, QWORD PTR [RIP + 0x1147A02]
MOV      R8D, QWORD PTR 0x[0x50]
MOV      RCX, QWORD PTR [RBP + R9 + 0x8]
JLEA     R11, QWORD PTR [RIP + 0x2A07D]
MOV      R10, 0xFFFFFFFFF2345A
ADD      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], R11
MOV      R11, 0xFFFFFFFFF5EAB
ADD      R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
JMP      R8D, QWORD PTR [RCX + 0x44]
CMOVLBE R11, R10
JLE      loc_182489f9d

loc_182489cee
LEA      RCX, QWORD PTR [RIP + 0x11E1E1E1]

```

```
MOV     RCX, 0x00000000000000B3
ADD     R10, QWORD PTR [RBP + 0x0000000000000000]
LEAQ    R12, QWORD PTR [RBP + 0x0000000000000000]
MOV     QWORD PTR [RBP + 0x0000000000000000], R10
JMOV    QWORD PTR [RBP + 0x0000000000000000], RCX
R11D, 0x00000000
ADD     R11, QWORD PTR [RBP + 0x0000000000000000]
JMOV    RCX, 0x0000000000000066
ADD     R10, QWORD PTR [RBP + 0x0000000000000000]
JMOV    QWORD PTR [RBP + 0x0000000000000000], R11
LEAQ    R11, QWORD PTR [RIP + 0x117250]
JMOV    QWORD PTR [RBP + 0x0000000000000000], R11
R11, 0x0000000000000000
ADD     R11, QWORD PTR [RBP + 0x0000000000000000]
JMOV    RB, 0x00000000000000A7
ADD     R8, QWORD PTR [RBP + 0x0000000000000000]
JMOV    QWORD PTR [RBP + 0x0000000000000000], R8
MOV     R12, QWORD PTR [RBP + 0x0000000000000000]
JCALL  R11

```

↓

loc_16248d58

RBP	R0V	RAX	R12
-----	-----	-----	-----

```
MOV     RCX, 0x0FFFFFFF0000005555
ADD     RDX, RCX
JMOV    R8D, QWORD PTR [RBP + 0x0FFFFFFF0000005555]
JLEA    R11, QWORD PTR [R12 + 0x0FFFFFFF0000005555]
JMOV    QWORD PTR [RBP + 0x0FFFFFFF0000005555], R11
ADD     R11, 0xFFFFFFFFFA900
MOV     R11, QWORD PTR [RBP + 0x0FFFFFFF0000005555]
JMP     R8D, 0xFFFFFFFF
CMOVNE  R11, RCX
JNZ     loc_182489f9d
```

```

LEA     RCX, QWORD PTR [RIP + 0x120456]
[MOV    RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFF0], RCX
MOV     RCX, 0xFFFFFFFF45028
ADD     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
[MOV    RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], RCX
MOV     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
[MOV    RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFD098]
[LEA     R11, QWORD PTR [RIP + 0xFFFFFFFFF637119]
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFF0], R11
[MOV    R11D, 0x50FD2
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFF0]
[MOV    RCX, QWORD PTR [0x7FED265]
XOR     RAX, RAX
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFF8], RAX
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFF8], R11
MOV     R11, 0xFFFFFFFF79BC
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFF8]
[LEA     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFF8]
[MOV    RCX, R12
CALL    R11

```

```

lea     rcx, qword ptr [rip + 0x0000000000000009]
mov     qword ptr [rbp + 0x0000000000000000], rcx
mov     rcx, 0x0000000000000543
add     rcx, qword ptr [rbp + 0x0000000000000000]
mov     rcx, qword ptr [rbp + 0x0000000000000008], rcx
mov     r11, 0x00000000000000a6
add     r11, qword ptr [rbp + 0x0000000000000008]
mov     r10, qword ptr [rbp + 0x0000000000000008], 0x0
lea     rcx, qword ptr [rbp + 0x0000000000000008]
mov     r12, r10
call    r11

```

```

MOV     RCX, QWORD PTR [RBP + 0x0000000000000000]
[SHLD]    RCX, RCX, 0x20
MOV     QWORD PTR [RBP + 0x0000000000000000], RCX
MOV     R13, 0xC0D5171F96037A1F
ADD     RCX, R13
RBDI, EDI
MOV     RCX, RCX
RAX, RCX
MOV     R11, RCX
R9, RDX
MOV     BYTE PTR [RBP + 0x0000000000000000], DH
MOV     BYTE PTR [RBP + 0x0000000000000000], DL
R10D, EDX
R10D, 0x10
SHR     R8D, 0x18
SHR     RCX, 0x20
SHR     RAX, 0x28
SHR     R11, 0x30
SHR     R9, 0x38
MOV     BYTE PTR [RBP + 0x0000000000000000], R8B
MOV     BYTE PTR [RBP + 0x0000000000000000], R10B
MOV     BYTE PTR [RBP + 0x0000000000000000], AL
MOV     BYTE PTR [RBP + 0x0000000000000000], CL
MOV     BYTE PTR [RBP + 0x0000000000000000], R9B
MOV     RCX, QWORD PTR [RBP + 0x0000000000000000]
[SHLD]    RCX, RCX, 0x8
MOV     EAX, DWORD PTR [RBP + 0x0000000000000000]
[XOR]     RCX, RCX
LEA     RCX, QWORD PTR [RIP + 0x0000000000000000]
[MOV]     QWORD PTR [RBP + 0x0000000000000000], RCX
EXCH, 0xB3E1F
ADD     RCX, QWORD PTR [RBP + 0x0000000000000000]
[NOT]     RAX
MOV     QWORD PTR [RBP + 0x0000000000000000], RCX
MOV     RCX, RAX
SHL     RCX, 0x1F
SUB     RAX, RCX
MOV     QWORD PTR [RBP + 0x0000000000000000], RAX
R11, 0x0000000000000022F
ADD     R11, QWORD PTR [RBP + 0x0000000000000000]
[MOV]     RCX, R12
CALL     R11

```

```

loc_182489dce
MOV     ECX, QWORD PTR [RBP + 0x00000000FFFFFFF8]
JMOV    R11, RCX
SHL     R11, 0x8
XOR     R11, RCX
MOV     RCX, R11
SHL     RCX, 0x16
MOV     R8, R11
SHL     R8, 0x2C
XOR     RCX, R8
XOR     R11, R8
SHLD    R8, R8, 0x20
MOV     ECX, DWORD PTR [0x7FFB0330]
JLEA    R11, QWORD PTR [RIP + 0x0000000048CA5A]
JMOV    QWORD PTR [RBP + 0x00000000FFFFFFF0], R11
MOV     R11, 0x00000000FFC4DE2
ADD     R11, QWORD PTR [RBP + 0x00000000FFFFFFF0]
JXOR    RCX, R8
MOV     QWORD PTR [RBP + 0x00000000FFFFFFF8], R11
BSWAP   RCX
MOV     QWORD PTR [RBP + 0x00000000FFFFFFF8], RCX
MOV     R11, 0x00000000FFC0305
ADD     R11, QWORD PTR [RBP + 0x00000000FFFFFFF8]
JMOV    RCX, R12

```

The diagram illustrates the sequential execution of assembly instructions. A vertical blue arrow points downwards through the center of the page, starting from the first instruction and ending at the last one. Each instruction is presented as a pair of columns: the instruction name on the left and its full assembly syntax on the right.

	loc_18248a0ab
MOV	RCX, QWORD PTR [RBP + 0x7FFFFFFFEEFE8]
[BSWAP	RCX
MOV	EAX, DWORD PTR [0x7FF50130]
[XOR	RAX, RCX
LEA	RCX, QWORD PTR [RIP + 0x0DEB39]
[MOV	QWORD PTR [RBP + 0x7FFFFFFFEEFE8], RCX
SHLD	RAX, RAX, QWORD PTR [RBP + 0x7FFFFFFFEEFE8], RCX
MOV	RCX, RAX
SHL	RCX, RCX, 8
MOV	R11, 0xFFFFFFFFFFFF5960
ADD	R11, QWORD PTR [RBP + 0x7FFFFFFFEEFE8], R11
[MOV	QWORD PTR [RBP + 0x7FFFFFFFEEFE8], R11
XOR	RCX, RAX
MOV	QWORD PTR [RBP + 0x7FFFFFFFEEFE8], RCX
MOV	R11, 0xFFFFFFFFFFFF5DB05
ADD	R11, QWORD PTR [RBP + 0x7FFFFFFFEEFE8], R11
[MOV	RCX, R12
CALL	R11

```

MOV     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], DH
INVL     RCX, R11
MOV     EDI, QWORD PTR [0x7FFFE0330]
XOR     RDX, RCX
RDX
SHLD     RDX, RDX, cx38
MOV     R8D, EDI
MOV     RCX, RDX
MOV     RAX, RDX
MOV     R11, RDX
R9
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF8], DH
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF9], DL
R10D, EDI
SHR     R10D, 0x10
SHR     R8D, 0x18
SHR     RAX, 0x20
SHR     R11, 0x30
SHR     R9, 0x38
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFA], R8B
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFB], R10B
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFC], AL
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFD], CL
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFE], R9B
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF7], R11B
MOV     RCX, 0x329A8B069C85E1
ADD     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
JMOV     RAX, RCX
SHR     RAX, 0x20
MOV     R11, RCX
WORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], AX
WORD PTR [RBP + 0xFFFFFFFFFFFFFFFC], CX
R8D, ECX
SHR     R8D, 0x10
SHR     R11, 0x30

```

```

LEA     RCX, QWORD PTR [RIP + 0x4049D]
|MOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], RCX
LEA     RCX, QWORD PTR [RIP + 0x77F31]
|MOV    R11, 0xFFFFFFFFFBD8B
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
|MOV    QWORD PTR [RSP + 0xFFFFFFFFFFFFFFFF], RCX
MOV     RCX, 0xFFFFFFFFFBD8167
ADD     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
|CMP     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], 0x0
CMOVNZ  RCX, R11
MOV     R15, 0x0F0F0F0F0F0F0F0F
MOV     R14, 0x8968E8F81544777B
MOV     RDI, 0x317C37A879C11298

```

```
MOV     RCX, 0x0000000000000000
ADD     QWORD PTR [RBP + 0x0000000000000000], RCX
JMOV    QWORD PTR [RBP + 0x0000000000000000], RCX
MOV     WORD PTR [RBP + 0x0000000000000000], R11W
MOV     MOV     QWORD PTR [RBP + 0x0000000000000000], R8W
R11, 0x0000000000000000
ADD     R11, QWORD PTR [RBP + 0x0000000000000000]
JMOV    RCX, R12
CALL    R11

loc_182489efb
LEA     RCX, QWORD PTR [RIP + 0x1216B49]
JMOV    QWORD PTR [RBP + 0x0000000000000000], RCX
MOV     RCX, 0x0000000000000001
ADD     QWORD PTR [RBP + 0x0000000000000000], RCX
JMOV    EAX, DWORD PTR [0x000000260]
MOV     QWORD PTR [RBP + 0x0000000000000000], RCX
R11, QWORD PTR [RBP + 0x0000000000000000]
JMOV    QWORD PTR [RBP + 0x0000000000000000], R11
LEA     RCX, QWORD PTR [RIP + 0x0000000000000000]
R12, QWORD PTR [RIP + 0x0018E398]
MOV     QWORD PTR [RBP + 0x0000000000000000], RCX
JMOV    ECX, 0x0F390
ADD     RCX, QWORD PTR [RBP + 0x0000000000000000]
JMOV    QWORD PTR [RBP + 0x0000000000000000], RCX
LEA     RCX, QWORD PTR [RIP + 0x0019CD01]
JMOV    QWORD PTR [RBP + 0x0000000000000000], RCX
MOV     RCX, 0x0000000000000A0A
ADD     RCX, QWORD PTR [RBP + 0x0000000000000000]
JMOV    R8D, 0x8A51E
ADD     R8, QWORD PTR [RBP + 0x0000000000000000]
JMOV    QWORD PTR [R11 + 0x0000000000000000], R8
JMOV    QWORD PTR [RBP + 0x0000000000000000], RAX
R11, 0x0000000000000A5E
ADD     R11, QWORD PTR [RBP + 0x0000000000000000]
JMOV    R8, QWORD PTR [RIP + 0x43CAB]
JMOV    QWORD PTR [RBP + 0x0000000000000000], R8
```

The diagram shows a jump instruction at memory location `loc_18248a1a4`. The instruction is `JMP RCX, QWORD PTR [RCX + 0x60]`. A red arrow points from the instruction to the target location `loc_18248a1a4`, which is also the location of the instruction itself, indicating a self-jump.

```

JMOV    QWORD PTR [RBP + 0x0FFFFFFFFFFFFFFF0], R8
MOV     R12, 0xFFFFFFFFFFFF4AAA
ADD     R12, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
JCALL   R11

```

```

1dc_182469f9a
JMP     R12

```

```
LEA    R11, QWORD PTR [RIP + 0x16993]
JMOV   QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFFF8], R11
MOV     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFFF8]
JLEA    R11, QWORD PTR [RIP + 0xC998B]
JMOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFFF0], R11
MOV     R11, 0xFFFFFFFFFFFF36F9A
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFFF0]
JMOV    RCX, QWORD PTR [RCX + 0xFFFFFFFFFFFF612AC]
JMP     loc_18248aab0
```

```

AND    R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
MOV     RCX, QWORD PTR [RCX + 0x18]
MOV     MOV     R12, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0], R11
R11, 0xFFFFFFFFFFFFFFF0
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
MOV     R12, QWORD PTR [RCX + 0x10]
ICALL

```

loc_18248a1e0

loc_182489f9a
JMP R12

```
LEA    R11, QWORD PTR [RIP + 0xFFFFFFFFF6B795C]
MOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], R11
MOV    R11, 0xFFFFFFFFF6B79576
ADD    R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
JMOV   RAX, DWORD PTR [0x7FFED260]
JXOR   RAX, RCX
```

```
MOVXZ    ECX, WORD PTR [R12 + 0x58]
MOV      QWORD PTR [RBP + 0x0000000000000000], R14
LEA      R9, QWORD PTR [RIP + 0x11a2694]
MOV      QWORD PTR [RBP + 0x0000000000000000], R9
MOV      R8D, 0x00000000
ADD      QWORD PTR [RBP + 0x0000000000000000], R10
R10 = 0x01007525d1076935
MOV      QWORD PTR [RBP + 0x0000000000000000], R10
R10 = 0x0000000000000000
MOV      QWORD PTR [RBP + 0x0000000000000000], R8
MOV      R10, QWORD PTR [RBP + 0x0000000000000000]
MOV      QWORD PTR [RBP + 0x0000000000000000], R10
BSWAP    R10
MOV      R9, R10
SHR      R9, 0x4
AND      R9, R15
AND      R10, R15
SHL      R10, 0x4
OR       R10, R9
MOV      R9, R10
MOV      RAX, 0x3333333333333333
AND      RAX, RAX
SHR      R9, 0x2
```

```
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], RAX
MOV     R11, 0xFFFFFFFFFFFF8BA7
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
JMOV    RCX, R12
CALL    R11
```

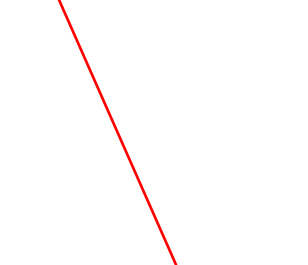
```

LEA    R10, QWORD PTR [R10 + R9 * 0x4]
|MOV   R9, R10
MOV    RAX, 0x5555555555555555
AND    R9, RAX
SHR    R10, 0x1
AND    R10, RAX
LEA    R10, QWORD PTR [R10 + R9 * 0x2]
|MOV   R8, QWORD PTR [R8 + 0x4ffffff39da]
|CMP   R10, R8
SETZ   BYTE PTR [RBP + 0xffffffffffffe8]
|MOVZX EAX, BYTE PTR [RBP + 0xffffffffffffe8]
|LEA   R8, QWORD PTR [RIP + 0x719a8]
|MOV   QWORD PTR [RBP + 0xfffffffffffff8], R8
LEA    R8, QWORD PTR [RIP + 0x8b6a4]
|MOV   R10, 0xffffffff86af
ADD    R10, QWORD PTR [RBP + 0xfffffffffffff8]
|MOV   QWORD PTR [RBP + 0xfffffffffffff8], R8
ADD    R8, 0xffffffff7979
MOV    R8, QWORD PTR [RBP + 0xfffffffffffff8]
|TEST  AL, 0x1
CMOVNZ CMVNZ, R8, R10
JNZ    loc_18248a2da

```

```
MOV     RCX, RDX
MOV     RAX, RDX
MOV     R11, RDX
MOV     R9, RDX
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF8], DH
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF9], DL
MOV     R10D, EDI
MOV     R10D, 0x10
SHR     R8D, 0x18
SHR     RCX, 0x20
SHR     RAX, 0x28
SHR     R11, 0x30
SHR     R9, 0x38
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFA], R8B
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFB], R10B
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFC], AL
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFFD], CL
```

```
loc_18248a2da
MOVEX    EAX, BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF8]
LEA      R9, QWORD PTR [RIP + 0x7a850]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], R9
LEA      R8, QWORD PTR [RIP + 0xF9896]
MOV      R10, 0xFFFFFFFFFFFF570E
ADD      R10, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
MOV      QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], R8
```



```
LEA    R8, QWORD PTR [RIP + 0x52A3F]
JMOV   QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], R8
LEA    R8, QWORD PTR [RIP + 0x417C1]
JMOV   QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], R8
MOV     R8, 0xFFFFFFFFFFFFFFFFB8556
```

```
MOV     BYTE PTR [RBP + 0xFFFFFFFFFFFFFFF0], R11B
MOV     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
JSHLD   RCX, RCX, 0x8
MOV     EAX, DWORD PTR [0x7FE0330]
JXOR    RAX, RCX
LEA     RCX, QWORD PTR [RIP + 0xFFFFFFFFFEE5C8CC]
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0], RCX
RCX,    0xFFFFFFFFFFFFFF260C
ADD     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
JNOT    RAX
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], RCX
MOV     RCX, RAX
SHL     RCX, 0x1F
SUB     RAX, RCX
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], RAX
R11,    0xFFFFFFFFFFFFFFD0FF
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
JND     RCX, R12
```

```
ADD      R8, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
JTEST    AL, 0x1
CMOVNZ   R8, R10
JMP
```

```
]MOV     R10, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
]RETF
```

```

MOV     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFF8]
JBSKAP  EAX
MOV     ECX, DWORD PTR [0x7FFE0330]
JXOR    RAX, RCX
LEA     RCX, QWORD PTR [RIP + 0xC86CFB]
JXOR    QWORD PTR [RBP + 0xFFFFFFFFFFFFF0], RCX
SHLD    RAX, RAX, 0x20
MOV     RCX, RAX
MOV     R11D, 0x1451D
ADD     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFF0]
JSHL    R11, RCX, 0xB
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFF8], R11
XOR     RCX, RAX
MOV     QWORD PTR [RBP + 0xFFFFFFFFFFFFF8], RCX
MOV     R11, 0xFFFFFFFFF5C2

```

```

1JEA    RCX, QWORD PTR [RIP + 0x12132AC]
1JMOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0], RCX
1MOV     R13, 0xFFFFFFFFFFFFFF3B44h
1ADD     R13, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF0]
1JMOV    RCX, R12
1CALL   R11

```

↓

```

loc_18248abf2
1MOV     RSI, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF5]
1JMOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], R13
1MOV     RCX, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8]
1RSD, DWORD PTR [RCX + 0xFFFFFFFFFFFFFFB4C]
1JEA     RCX, QWORD PTR [RIP + 0x112A1]
1JMOV    QWORD PTR [RBP + 0xFFFFFFFFFFFFFFF8], RCX
1RSD, DWORD PTR [RIP + 0x11496D]
1JMOV    RCX, QWORD PTR GS:[0x58]

```

```
LEA    R10, QWORD PTR [RIP + 0x516A4]
JMOV   R10, 0xFFFFFFFFFFFFE07D
ADD     R10, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
JMOV   R10, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF], R11
MOV     R11, 0xFFFFFFFFFFFFA998
ADD     R11, QWORD PTR [RBP + 0xFFFFFFFFFFFFFFFF]
JCMPEQ R8D, DWORD PTR [RCX + 0x44]
JCMOVLE R11, R10
MOV     RBX, 0xFAF89BCD2E1763209
MOV     R13, 0x3333333333333333
JMP     R11
```