0x2a11

if (\$s14) goto 0x2a25

rets \$s11

```
0x2a25
                                                                                                                                                                     -----
                                                                                                                                                                     assert(s[0x7] < block.number)</pre>
                                                                                                                                                                     $s18 = 0 == (ad_mask \& $s13)
                                                                                                                                                                                                           0x2a1d
                                                                                                                                                                     if (! $s18){
                                                                                                                                                                                                            -----
                                                                                                                                                                      $s18 = (ad mask \& $s13) == self
                                                                                                                                                                                                           $s15 = 0x1
                                                                                                                                                                                                           goto 0xe7e
                                                                                                                                                                     assert(0 == $s18)
                                                                                                                                                                     $s18 = intcall14(block.number, $s12, 0x2a71)
                                                                                                                                                                     $s16 = $s18
                                                                                                                                                                     if ($s18 >= $s14) goto 0x2a84
0x2a84
$s18 = intcall21(ad_mask & s[0x1], 0x2a99)
if ($s18){
 m[0x20 + \$m] = 0x0
 m[0x4 + $m] = ad_mask \& $s12

m[0x24 + $m] = ad_mask \& $s13

m[0x44 + $m] = $s14
 $s22 = ad_mask \& s[0x1]
 assert(ex\overline{t}codesize($s22))
 assert(call(msg.gas - 0x2c6, $s22, 0x0, $m, 0x64, $m, 0x20))
 assert(m[$m])
m[0x0] = ad_mask \& $s12
m[0x20] = 0\overline{x}9
$s18 = sha3(0x0, 0x40)
$s20 = intcall19($s14, $s16, 0x184d)
$s23 = 0 == s[$s18]
if (! $s23){
 t = s[$s18]
 $s25 = $t - 0x1
 assert($s25 < $t)
 m[0x0] = $s18
 if ($s23){
 $s23 = s[$s18]
 $s27 = 0x1 + $s23
 $s28 = s[$s18]
 s[\$s18] = \$s27
 if (0 == $s28 <= $s27){
   m[0x0] = $s18
   $s29 = sha3(0x0, 0x20)
                                                                                                                                                                                                   0x2a7c
   t = s29
   $s29 = $s29 + $s28
                                                                                                                                                                                                   $s15 = 0x0
   $s31 = $s27 + $t
                                                                                                                                                                                                   goto 0xe7e
   while (0x1) {
    if ($s29 <= $s31)
      break
     s[\$s31] = 0x0
     $s31 = 0x1 + $s31
   goto 0x1e7d
 assert($s23 < s[$s18])
 m[0x0] = \$s18
 $s23 = $s23 + sha3(0x0, 0x20)
 goto 0x2b6f
 } else {
 $s23 = s[$s18]
$s24 = $s23 - 0x1
 assert($s24 < $s23)
 m[0x0] = $s18
 $s23 = $s24 + sha3(0x0, 0x20)
 $s18 = intcall14(block.number, $s13, 0x2b79)
m[0x0] = ad_{mask} \& $s13
m[0x20] = 0x9
= intcall20($s14, $s18, 0x184d, sha3(0x0, 0x40), 0x2bad)
$s21 = m[0x0]
codecopy(0x0, 0x2d0c, 0x20)

$s20 = m[0x0]

m[0x0] = $s21
m[\$m] = \$\$14
log3(\$m, (0x20 + \$m) - \$m, \$s20, ad\_mask \& \$s12, ad\_mask \& \$s13)
goto $s11
                                                                                                                                                                                                       0xe7e
                                                                                                                                                                                                        -----
                                                                                                                                                                                                       $s11 = $s15
                                                                                                                                                                                                       intret()
```