

Untitled

BY: A GUEST ON SEP 4TH, 2015 | SYNTAX: NONE | SIZE: 4.99 KB | VIEWS: 71 | EXPIRES: NEVER

DOWNLOAD | RAW | EMBED | REPORT ABUSE | PRINT | QR CODE | CLONE

ADS VIA

Carbor

0 Public P

0

Untit2 sec

Untit3 sec

Twitt11 se

My St18 se

Untit17 se

Untit35 se

Untit44 se

i mes53 se

```
1.
2. #!/usr/bin/env python2.7
3. #By aaaddress1@gmail.com
4.
5. from z3 import *
6. b = [ BitVec('b%d' % i, 32) for i in range(16) ]
7. s = Solver()
8.
9. for i in range(16):
10.     s.add(b[i] & 0xff == b[i])
11.
12. # _ _ _
13. # | | | _ _ _ | _
14. # | | | / _ / _ ' _ \
15. # | _ | ( | \ \ | | |
16. # | | | \ _ _ | _ / | |
17. #
18.
19. v6 = BitVec('s1', 32)
20. v7 = BitVec('s2', 32)
21. v8 = BitVec('s3', 32)
22. v9 = BitVec('s4', 32)
23. s.add(v6 == 0 and v7 == 0 and v8 == 0 and v8 == 0)
24.
25. '''
26. v4 = 0LL;
27.     v6 = 0;
28.     v7 = 0;
29.     v8 = 0;
30.     v9 = 0;
31.     do
32.     {
33.         v10 = v1[v4];
34.         if ( (unsigned __int8)(v10 - 33) > 0x59u )
35.         {
36.             LODWORD(v4) = 0;
37.             return (unsigned int)v4;
38.         }
39.         v11 = __ROR4__(v8, 23);
40.         ++v4;
41.         v9 += v10;
42.         v7 = v10 ^ 8 * v7;
43.         v6 = v10 + 32 * v6;
44.         v8 = 37 * v10 + v11;
45.     }
46.     while ( v4 != 16 );
47. '''
48. for v4 in range(16):
49.     v10 = b[v4]
50.     s.add((0xff & (v10 - 33)) <= 0x59)
51.     v11 = RotateRight(v8, 23)
52.     v9 += v10
53.     v7 = v10 ^ 8 * v7
54.     v6 = v10 + 32 * v6
55.     v8 = 37 * v10 + v11
56. # if ( v9 == 1720 && v8 == 14470772 && v7 == -340847546 && v6 == -1721686258 )
57. s.add(v6 == 0x9961270E)
```

```

58. s.add(v7 == 0xEBAF1446)
59. s.add(v8 == 14470772)
60. s.add(v9 == 1720)
61.
62.
63. #   _ _ _
64. # / _ | _ _ _ | ( ) | _ ( ) _ _ _
65. # | | / _ \ | ' _ \ / _ | | _ | / _ \ | ' _ \
66. # | | _ ( ) | | | ( | | | _ | ( ) | | |
67. # \ _ _ \ / _ | _ | \ _ _ | _ | \ _ _ \ / _ | _ |
68. #
69. v = [0] * 16
70. for i in range(16):
71.     v[i] = b[i]
72. #((_DWORD *)v1 ^= 0x11111111u;
73. for i in range(0,4):
74.     v[i] = v[i] ^ 0x11;
75. #((_DWORD *)v1 + 1) ^= 0xAAAAAA;
76. for i in range(4,8):
77.     v[i] = v[i] ^ 0xAA
78. #((_DWORD *)v1 + 2) ^= 0x55555555u;
79. for i in range(8,12):
80.     v[i] = v[i] ^ 0x55
81. #((_DWORD *)v1 + 3) ^= 0x33333333u;
82. for i in range(12,16):
83.     v[i] = v[i] ^ 0x33
84. #v12 = v1[1] ^ *v1;
85. v12 = (0xff & v[1]) ^ v[0]
86. #v13 = (unsigned __int8)(v1[2] ^ v12);
87. v13 = 0xff & (v[2] ^ v12)
88. #v14 = (unsigned __int8)(v1[3] ^ v1[2] ^ v12);
89. v14 = 0xff & (v[3] ^ v[2] ^ v12)
90. #v15 = (unsigned __int8)(v1[4] ^ v1[3] ^ v1[2] ^ v12);
91. v15 = 0xff & (v[4] ^ v[3] ^ v[2] ^ v12)
92. #v16 = (unsigned __int8)(v1[5] ^ v1[4] ^ v1[3] ^ v1[2] ^ v12);
93. v16 = 0xff & (v[5] ^ v[4] ^ v[3] ^ v[2] ^ v12)
94. #v17 = (unsigned __int8)(v1[6] ^ v1[5] ^ v1[4] ^ v1[3] ^ v1[2] ^ v12);
95. v17 = 0xff & (v[6] ^ v[5] ^ v[4] ^ v[3] ^ v[2] ^ v12)
96. #v18 = v1[8] ^ v1[7] ^ v1[6] ^ v1[5] ^ v1[4] ^ v1[3] ^ v1[2] ^ v12;
97. v18 = (v[8] ^ v[7] ^ v[6] ^ v[5] ^ v[4] ^ v[3] ^ v[2] ^ v12)
98. #v19 = (unsigned __int8)(v1[7] ^ v1[6] ^ v1[5] ^ v1[4] ^ v1[3] ^ v1[2] ^ v12);
99. v19 = 0xff & (v[7] ^ v[6] ^ v[5] ^ v[4] ^ v[3] ^ v[2] ^ v12)
100. #((_BYTE *)v1 + 8) = v18;
101. v[8] = v18
102. #v20 = v1[9] ^ v18;
103. v20 = v[9] ^ v18
104. #v21 = (unsigned __int8)(v1[10] ^ v20);
105. v21 = 0xff & (v[10] ^ v20)
106. #v22 = (unsigned __int8)(v1[11] ^ v1[10] ^ v20);
107. v22 = 0xff & (v[11] ^ v[10] ^ v20)
108. #v23 = (unsigned __int8)(v1[12] ^ v1[11] ^ v1[10] ^ v20);
109. v23 = 0xff & (v[12] ^ v[11] ^ v[10] ^ v20)
110. #v24 = v1[14] ^ v1[13] ^ v1[12] ^ v1[11] ^ v1[10] ^ v20;
111. v24 = v[14] ^ v[13] ^ v[12] ^ v[11] ^ v[10] ^ v20
112. #v25 = v1[13] ^ v1[12] ^ v1[11] ^ v1[10] ^ v20;
113. v25 = v[13] ^ v[12] ^ v[11] ^ v[10] ^ v20
114. #((_BYTE *)v1 + 15) ^= v24;
115. v[15] ^= v24
116. #*v1 ^= 0x63u;
117. v[0] ^= 0x63
118. #((_BYTE *)v1 + 8) ^= 0x30u;
119. v[8] = 0x30
120. #((_BYTE *)v1 + 1) = (2 * v12 | ((signed int)v12 >> 1)) ^ 0x2F;
121. v[1] = (2 * v12 | v12 >> 1) ^ 0x2f
122. #((_BYTE *)v1 + 2) = (4 * v13 | (v13 >> 2)) ^ 0xDC;
123. v[2] = (4 * v13 | v13 >> 2) ^ 0xdc
124. #((_BYTE *)v1 + 3) = (8 * v14 | (v14 >> 3)) ^ 0x20;
125. v[3] = (8 * v14 | v14 >> 3) ^ 0x20
126. #((_BYTE *)v1 + 4) = (16 * v15 | (v15 >> 4)) ^ 0xCD;
127. v[4] = (16 * v15 | v15 >> 4) ^ 0xcd
128. #((_BYTE *)v1 + 5) = (32 * v16 | (v16 >> 5)) ^ 0xA0;
129. v[5] = (32 * v16 | v16 >> 5) ^ 0xa0
130. #((_BYTE *)v1 + 6) = (((_BYTE)v17 << 6) | (v17 >> 6)) ^ 0x83;
131. v[6] = ( (0xff & v17) << 6 | v17 >> 6) ^ 0x83

```

```
132. #*((_BYTE *)v1 + 7) = ((_BYTE)v19 << 7) | (v19 >> 7);
133. v[7] = ( (0xff & v19) << 7 | v19 >> 7)
134. #*((_BYTE *)v1 + 9) = (2 * v20 | ((signed int)v20 >> 1)) ^ 0x7D;
135. v[9] = (2 * v20 | v20 >> 1) ^ 0x7d
136. #*((_BYTE *)v1 + 10) = (4 * v21 | (v21 >> 2)) ^ 0x19;
137. v[10] = (4 * v21 | v21 >> 2) ^ 0x19
138. #*((_BYTE *)v1 + 11) = (8 * v22 | (v22 >> 3)) ^ 4;
139. v[11] = (8 * v22 | v22 >> 3) ^ 4
140. #*((_BYTE *)v1 + 12) = (16 * v23 | (v23 >> 4)) ^ 0xC4;
141. v[12] = (16 * v23 | v23 >> 4) ^ 0xc4
142. #*((_BYTE *)v1 + 13) = (32 * v25 | ((signed int)v25 >> 5)) ^ 0x20;
143. v[13] = (32 * v25 | v25 >> 5) ^ 0x20
144. #LODWORD(v4) = v1[15];
145. v4 = v[15]
146. #*((_BYTE *)v1 + 14) = ((v24 << 6) | ((signed int)v24 >> 6)) ^ 0xC1;
147. v[14] = (v24 << 6 | v24 >> 6) ^ 0xc1
148. #*((_BYTE *)v1 + 15) = ((_BYTE)v4 << 7) | ((signed int)v4 >> 7);
149. v[15] = (v4 << 7 | v4 >> 7)
150. #LODWORD(v4) = ((*(_DWORD *)v1 + 3) | ((*(_DWORD *)v1 + 2) | ((*(_DWORD *)v1 | ((*(_DWORD *)v1 + 1)) == 0;
151. s.add(v[0] & 0xff == 0)
152. s.add(v[1] & 0xff == 0)
153. s.add(v[2] & 0xff == 0)
154. s.add(v[3] & 0xff == 0)
155. s.add(v[4] & 0xff == 0)
156. s.add(v[5] & 0xff == 0)
157. s.add(v[6] & 0xff == 0)
158. print(s.check())
159.
160. m = s.model()
161. res = ""
162. for i in range(16):
163.     v = int(str(m[b[i]]))
164.     c = chr(v)
165.     print('%d: %d %s' % (i, v, c))
166.     res += c
167.
168. print("Key is %s" % res)
```

RAW Paste Data

[clone this paste](#)

```
#!/usr/bin/env python2.7
#By aaaddress1@gmail.com

from z3 import *
b = [ BitVec('b%d' % i, 32) for i in range(16) ]
s = Solver()

for i in range(16):
    s.add(b[i] & 0xff == b[i])

# _ _ _
# | | | _ _ | _
# | | | | | | |
```

Pastebin T-shirt Shop

high quality geeky t-shirts



Pastebin.com Tools & Applications

- iPhone/iPad
- Windows
- Firefox
- Chrome
- WebOS
- Android
- Mac
- Opera
- Click.to
- UNIX
- WinPhone

create new paste | api | trends | syntax languages | faq | tools | privacy | cookies | contact | dmca | advertise on pastebin | shop **NEW!** | go pro

Follow us: pastebin on facebook | pastebin on twitter | pastebin in the news

Dedicated Server Hosting by Steadfast

Pastebin v3.11 rendered in: 0.007 seconds

Site design & logo © 2015 Pastebin; user contributions (pastes) licensed under cc by-sa 3.0