Customization

date 24.10.2021

solved in time of CTF

category Reverse Engineering score 349

Description

Custom string encryption

Attached files

Disassembled class file

```java
public static String enc(char[] paramArrayOfchar, int paramInt) {
    String str1 = "";
    for (int b1 = 0; b1 < paramArrayOfchar.length; b1++)
        str1 = str1 + String.format("%08d",
Integer.valueOf(Integer.toBinaryString(paramArrayOfchar[b1])));
    while (str1.length() % 6 != 0)
        str1 = str1 + "0";
    String str2 = "";
    for (int b2 = 0; b2 < str1.length() / 6; b2++)
        str2 = str2 + x[Integer.parseInt(str1.substring(b2 * 6, b2 * 6 + 6), 2)];
    if (paramInt > 1)
        return enc(str2.toCharArray(), paramInt - 1);
    return str2;
}
static char[] x = new char[] {
    '"', 'd', 's', 'V', 'U', '8', 'q', ';', '|', 'G',
    'k', 'M', '9', 'K', '<', '2', 'C', 'a', 'c', ')',
    '5', 'v', 'D', ']', '7', '\'', '>', '$', 'y', 'E',
    'l', 'W', 'A', 'F', '%', 'i', 'O', '&', '*', '[',
    'Y', 'H', 'L', 'u', 'g', '(', 'B', ':', '?', '1',
    '!', '-', '0', 'w', 'J', 'R', 'm', 'f', 'N', '_',
    '{', 'b', 'p', '4'
};
```

solve.java

```java
static int[] s = new int[] {61, 41, 25, 102, 98, 115, 111, 26, 108, 127, 31, 56,
21, 107, 103, 24, 98, 3, 21, 102, 39, 119, 56, 12, 103, 8, 11, 12, 61, 55, 24, 40,
63, 110, 60, 34, 63, 26, 21, 22, 25, 5, 102, 24, 34, 3, 121, 53, 108, 102, 34, 5,
40, 26, 102, 20, 63, 110, 110, 56, 39, 107, 27, 1, 103, 116, 120, 20, 107, 116,
11, 41, 63, 61, 29, 127, 107, 116, 29, 51, 9, 8, 107, 41, 103, 26, 102, 55, 21,
115, 60, 19, 40, 119, 107, 7, 122, 47, 21, 34, 63, 55, 17, 1, 63, 107, 110, 46,
27, 115, 58, 122, 108, 55, 40, 102, 63, 47, 102, 122, 53, 102, 107, 56, 40, 119,
107, 7, 96, 20, 103, 51, 39, 3, 121, 22, 119, 41, 60, 43, 98, 47, 25, 21, 61, 110,
63, 37, 25, 55, 25, 115, 119, 127, 7, 56, 96, 3, 17, 18, 63, 12, 27, 20, 63, 5,
121, 55, 63, 5, 105, 127, 39, 119, 17, 18, 63, 107, 60, 8, 108, 110, 57, 22, 105,
17, 121, 43, 21, 61, 39, 51, 21, 12, 105, 12, 53, 17, 27, 26, 9, 45, 7, 56, 98,
11, 27, 41, 27, 123, 17, 116, 9, 45, 63, 40, 26, 40, 118, 24, 63, 47, 27, 12, 63,
40, 118, 5, 34, 3, 17, 41, 121, 127, 102, 101, 40, 5, 121, 53, 105, 119, 40, 56,
108, 110, 40, 108, 25, 115, 24, 43, 119, 12, 120, 118, 61, 41, 34, 5, 61, 17, 29,
24, 105, 40, 107, 37, 39, 61, 105, 1, 63, 12, 105, 116, 108, 115, 27, 53, 25, 55,
25, 11, 53, 3, 120, 34, 25, 115, 24, 101, 96, 61, 17, 97, 98, 40, 17, 127, 98, 8,
27, 118, 108, 45, 110, 111, 63, 47, 102, 20, 63, 110, 121, 56, 107, 5, 120, 103,
21, 127, 120, 20, 121, 123, 120, 40, 63, 107, 31, 5, 61, 123, 27, 53, 21, 115,
124, 41, 103, 26, 103, 24, 27, 110, 60, 101, 21, 107, 21, 47, 29, 115, 58, 58, 34,
119, 40, 108, 27, 17, 29, 12, 61, 123, 121, 108, 103, 17, 7, 56, 53, 11, 40, 108,
26, 3, 58, 19, 21, 3, 121, 55, 21, 20, 21, 12, 39, 119, 111, 40, 53, 11, 40, 101,
25, 17, 27, 47, 63, 40, 34, 111, 105, 41, 39, 51, 27, 115, 25, 24, 119, 110, 40,
43, 25, 12, 27, 37, 107, 17, 107, 97, 50, 115, 37, 37, 105, 12, 121, 61, 108, 8,
57, 5, 96, 26, 102, 20, 63, 110, 105, 56, 39, 110, 107, 18, 105, 127, 102, 108,
27, 115, 120, 40, 21, 61, 121, 102, 27, 20, 27, 12, 96, 127, 22, 102, 98, 119, 51,
118, 121, 41, 25, 11, 19, 61, 105, 118, 122, 11, 29, 97, 61, 47, 57, 22, 27, 123,
31, 56, 119, 41, 120, 116, 96, 41, 107, 7, 63, 26, 97, 41, 63, 17, 118, 27, 53,
40, 40, 121, 21, 12, 27, 101, 121, 116, 118, 22, 103, 3, 121, 102, 119, 110, 102,
122, 108, 45, 41, 123, 103, 116, 111, 55, 103, 45, 40, 101, 25, 17, 40, 11, 63,
107, 37, 12, 107, 55, 17, 1, 119, 127, 22, 19, 19, 119, 40, 122, 105, 40, 118,
108, 63, 116, 102, 27, 21, 61, 102, 56, 21, 26, 27, 122, 105, 40, 121, 20, 61, 55,
120, 108, 21, 61, 105, 12, 53, 107, 121, 41, 63, 41, 110, 116, 53, 101, 121, 61,
98, 11, 110, 37, 39, 119, 120, 118, 63, 119, 58, 20, 119, 107, 41, 11, 63, 12,
105, 12, 103, 47, 25, 122, 9, 45, 120, 102, 96, 61, 120, 53, 27, 20, 17, 22, 34,
107, 121, 34, 21, 127, 120, 20, 61, 120, 41, 27, 25, 110, 22, 19, 29, 97};
public static void main(String[] args) {
    char[] c = new char[s.length];
    for(int i = 0; i < s.length; i++)
        c[i] = (char) (((char) s[i]) ^ 0x5E);
    System.out.println(dec(c, 10));
}
public static String dec(char[] d, int n) {
    String str = "";
    for(int i = 0; i < d.length; i++)
        str += String.format("%06d",
Integer.valueOf(Integer.toBinaryString(indexof(d[i]))));
    if(str.length() % 8 != 0) {
        int fit = (str.length() / 8) * 8;
        str = str.substring(0, fit);
    }
    String str2 = "";
    for(int i = 0; i < str.length() / 8; i++)
        str2 += (char) Integer.parseInt(str.substring(i * 8, i * 8 + 8), 2);
    if(n > 1) return dec(str2.toCharArray(), n - 1);
    else return str2;
}
public static int indexof(char c) {
     for(int i = 0; i < x.length; i++)
         if(x[i] == c)
             return i;
     return -1;
 }
```

Summary

Straight forward character to binary then shuffled witch custom padding, turned back to char, and finally xor'ed.

Flag

```
hology4{JuSt_4_cUst0m_b4se64_W1th0ut_p4dd1ng}
```

Detailed solution

1. XOR the input string, this was fairly straight forward
2. Convert each characters to binary with length of 6.
3. Remove end padding
4. Reconvert the 6 length-ed binary string to 8 length-ed binary string.
5. Redo the step `n` times, in this case 10.
6. Done!

Another solutions

Bruteforce, unless you have a gigantic sized computer.