

Memecat BattleStation (FlareOn CTF)

Description:

Welcome to the Sixth Flare-On Challenge!

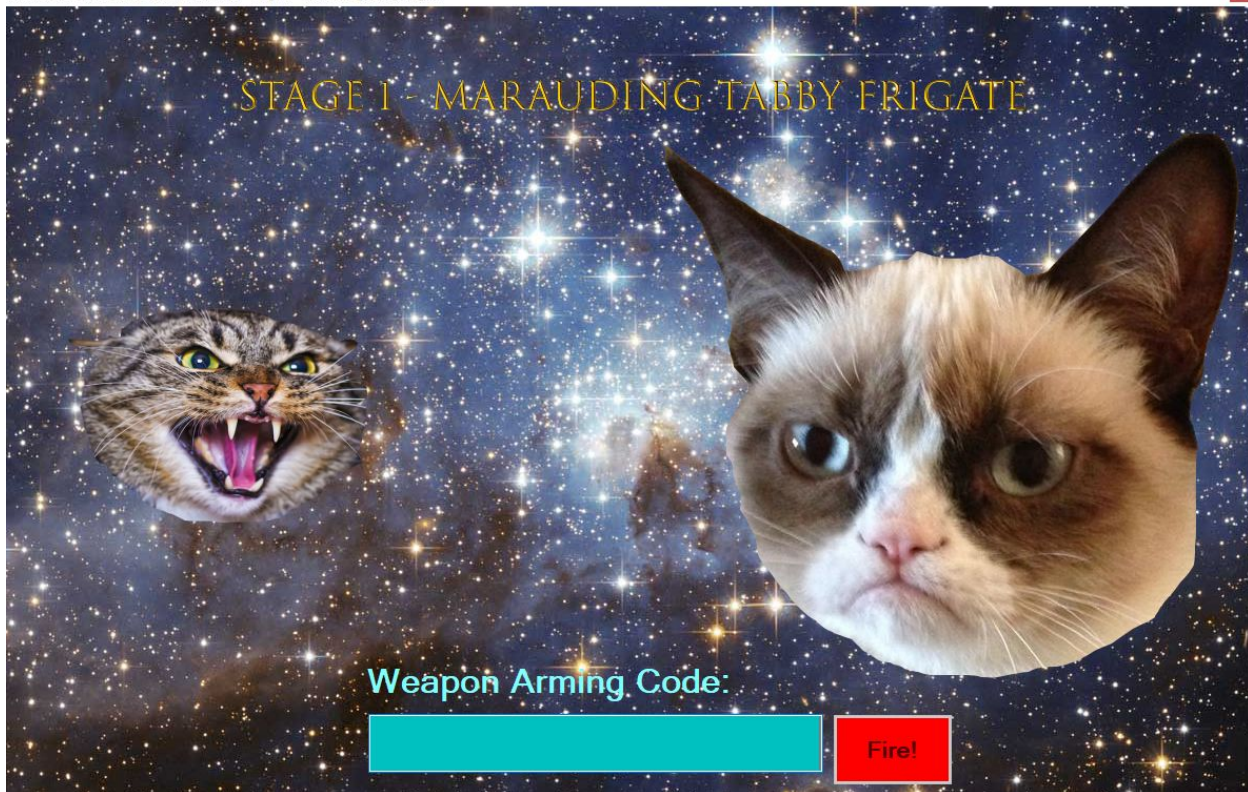
This is a simple game. Reverse engineer it to figure out what "weapon codes" you need to enter to defeat each of the two enemies and the victory screen will reveal the flag. Enter the flag here on this site to score and move on to the next level.

* This challenge is written in .NET. If you don't already have a favorite .NET reverse engineering tool I recommend dnSpy

** If you already solved the full version of this game at our booth at BlackHat or the subsequent release on twitter, congratulations, enter the flag from the victory screen now to bypass this level.

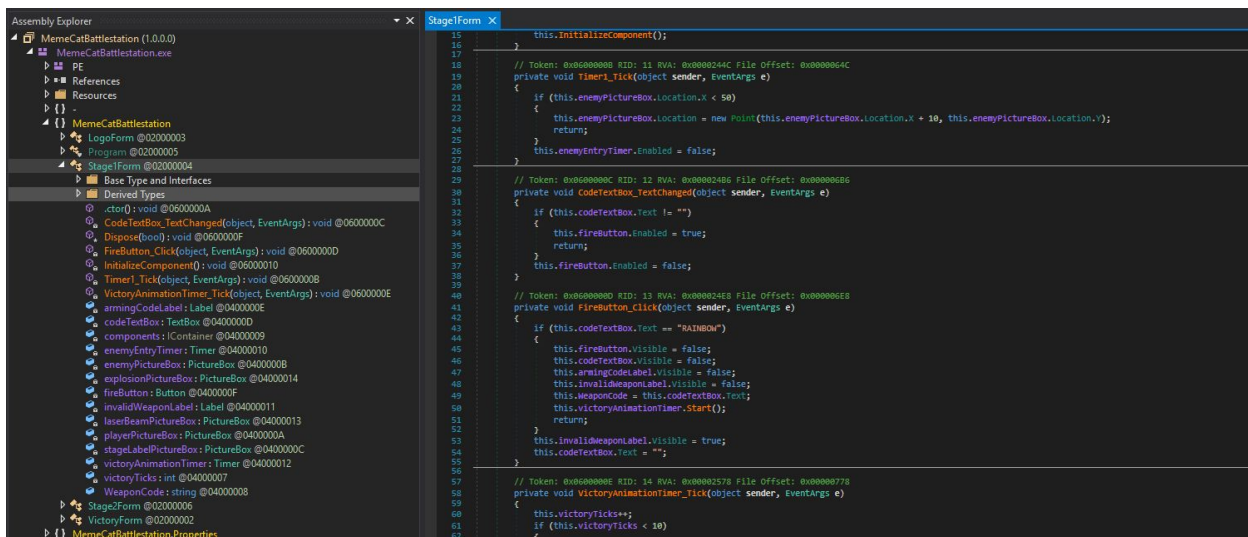
Challenge:

Running given exe file MemeCatBattlestation.exe shows Stage 1 Weapon Arming Code



As mentioned in the challenge description we can simply start reversing this binary with .Net decompilers. My first pick will be dnSpy.

Opening this exe in dnSpy shows **Stage1Form** which had below code snippet



We can see function **FireButton_Click** which contains string **RAINBOW**. Typing that in input field redirects us to Stage 2



Now checking **Stage2Form** shows another interesting function **IsValidWeaponCode()**

```
Assembly Explorer
MemeCatBattlestation (1.0.0.0)
  MemeCatBattlestation.exe
    PE
    References
    Resources
    {}
    {}
    MemeCatBattlestation
      LogoForm @02000003
      Program @02000005
      Stage1Form @02000004
      Stage2Form @02000006
        Base Type and Interfaces
        Derived Types
        .ctor(): void @06000012
        CodeTextBox_TextChanged(object sender, EventArgs e): void @06000015
        Dispose(bool): void @06000018
        EnemyEntryTimer_Tick(object sender, EventArgs e): void @06000013
        FireButton_Click(object sender, EventArgs e): void @06000017
        InitializeComponent(): void @06000019
        IsValidWeaponCode(string s): bool @06000016
        VictoryAnimationTimer_Tick(object sender, EventArgs e): void @06000014
        armingCodeLabel: Label @0400001A
        bagelsPictureBox: PictureBox @04000021
        codeTextBox: TextBox @04000018
        components: IContainer @04000017
        enemyEntryTimer: Timer @0400001F
        enemyPictureBox: PictureBox @0400001D

Stage2Form
67 private void CodeTextBox_TextChanged(object sender, EventArgs e)
68 {
69     if (this.codeTextBox.Text != "")
70     {
71         this.fireButton.Enabled = true;
72         return;
73     }
74     this.fireButton.Enabled = false;
75 }
76
77 // Token: 0x00000016 RID: 22 RVA: 0x000030C4 File Offset: 0x000012C4
78 private bool IsValidWeaponCode(string s)
79 {
80     char[] array = s.ToCharArray();
81     int length = s.Length;
82     for (int i = 0; i < length; i++)
83     {
84         char[] array2 = array;
85         int num = i;
86         array2[num] ^= 'A';
87     }
88     return array.SequenceEqual(new char[]
89     {
90         '\u0003',
91         '\u0004',
92         '\u0005',
93         '\u0006',
94         '\u0007',
95         '\u0008',
96         '\u0009',
97         '\u000A',
98         '\u000B',
99         '\u000C',
100        '\u000D',
101        '\u000E',
102        '\u000F',
103        '\u0010',
104        '\u0011',
105        '\u0012',
106        '\u0013',
107        '\u0014',
108        '\u0015',
109        '\u0016',
110        '\u0017',
111        '\u0018',
112        '\u0019',
113        '\u001A',
114        '\u001B',
115        '\u001C',
116        '\u001D',
117        '\u001E',
118        '\u001F',
119        '\u0020',
120        '\u0021',
121        '\u0022',
122        '\u0023',
123        '\u0024',
124        '\u0025',
125        '\u0026',
126        '\u0027',
127        '\u0028',
128        '\u0029',
129        '\u002A',
130        '\u002B',
131        '\u002C',
132        '\u002D',
133        '\u002E',
134        '\u002F',
135        '\u0030',
136        '\u0031',
137        '\u0032',
138        '\u0033',
139        '\u0034',
140        '\u0035',
141        '\u0036',
142        '\u0037',
143        '\u0038',
144        '\u0039',
145        '\u003A',
146        '\u003B',
147        '\u003C',
148        '\u003D',
149        '\u003E',
150        '\u003F',
151        '\u0040',
152        '\u0041',
153        '\u0042',
154        '\u0043',
155        '\u0044',
156        '\u0045',
157        '\u0046',
158        '\u0047',
159        '\u0048',
160        '\u0049',
161        '\u004A',
162        '\u004B',
163        '\u004C',
164        '\u004D',
165        '\u004E',
166        '\u004F',
167        '\u0050',
168        '\u0051',
169        '\u0052',
170        '\u0053',
171        '\u0054',
172        '\u0055',
173        '\u0056',
174        '\u0057',
175        '\u0058',
176        '\u0059',
177        '\u005A',
178        '\u005B',
179        '\u005C',
180        '\u005D',
181        '\u005E',
182        '\u005F',
183        '\u0060',
184        '\u0061',
185        '\u0062',
186        '\u0063',
187        '\u0064',
188        '\u0065',
189        '\u0066',
190        '\u0067',
191        '\u0068',
192        '\u0069',
193        '\u006A',
194        '\u006B',
195        '\u006C',
196        '\u006D',
197        '\u006E',
198        '\u006F',
199        '\u0070',
200        '\u0071',
201        '\u0072',
202        '\u0073',
203        '\u0074',
204        '\u0075',
205        '\u0076',
206        '\u0077',
207        '\u0078',
208        '\u0079',
209        '\u007A',
210        '\u007B',
211        '\u007C',
212        '\u007D',
213        '\u007E',
214        '\u007F',
215        '\u0080',
216        '\u0081',
217        '\u0082',
218        '\u0083',
219        '\u0084',
220        '\u0085',
221        '\u0086',
222        '\u0087',
223        '\u0088',
224        '\u0089',
225        '\u008A',
226        '\u008B',
227        '\u008C',
228        '\u008D',
229        '\u008E',
230        '\u008F',
231        '\u0090',
232        '\u0091',
233        '\u0092',
234        '\u0093',
235        '\u0094',
236        '\u0095',
237        '\u0096',
238        '\u0097',
239        '\u0098',
240        '\u0099',
241        '\u009A',
242        '\u009B',
243        '\u009C',
244        '\u009D',
245        '\u009E',
246        '\u009F',
247        '\u00A0',
248        '\u00A1',
249        '\u00A2',
250        '\u00A3',
251        '\u00A4',
252        '\u00A5',
253        '\u00A6',
254        '\u00A7',
255        '\u00A8',
256        '\u00A9',
257        '\u00AA',
258        '\u00AB',
259        '\u00AC',
260        '\u00AD',
261        '\u00AE',
262        '\u00AF',
263        '\u00B0',
264        '\u00B1',
265        '\u00B2',
266        '\u00B3',
267        '\u00B4',
268        '\u00B5',
269        '\u00B6',
270        '\u00B7',
271        '\u00B8',
272        '\u00B9',
273        '\u00BA',
274        '\u00BB',
275        '\u00BC',
276        '\u00BD',
277        '\u00BE',
278        '\u00BF',
279        '\u00C0',
280        '\u00C1',
281        '\u00C2',
282        '\u00C3',
283        '\u00C4',
284        '\u00C5',
285        '\u00C6',
286        '\u00C7',
287        '\u00C8',
288        '\u00C9',
289        '\u00CA',
290        '\u00CB',
291        '\u00CC',
292        '\u00CD',
293        '\u00CE',
294        '\u00CF',
295        '\u00D0',
296        '\u00D1',
297        '\u00D2',
298        '\u00D3',
299        '\u00D4',
300        '\u00D5',
301        '\u00D6',
302        '\u00D7',
303        '\u00D8',
304        '\u00D9',
305        '\u00DA',
306        '\u00DB',
307        '\u00DC',
308        '\u00DD',
309        '\u00DE',
310        '\u00DF',
311        '\u00E0',
312        '\u00E1',
313        '\u00E2',
314        '\u00E3',
315        '\u00E4',
316        '\u00E5',
317        '\u00E6',
318        '\u00E7',
319        '\u00E8',
320        '\u00E9',
321        '\u00EA',
322        '\u00EB',
323        '\u00EC',
324        '\u00ED',
325        '\u00EE',
326        '\u00EF',
327        '\u00F0',
328        '\u00F1',
329        '\u00F2',
330        '\u00F3',
331        '\u00F4',
332        '\u00F5',
333        '\u00F6',
334        '\u00F7',
335        '\u00F8',
336        '\u00F9',
337        '\u00FA',
338        '\u00FB',
339        '\u00FC',
340        '\u00FD',
341        '\u00FE',
342        '\u00FF'
343    });
344 }
```

This code is straightforward it just takes an array and does XOR operation with A.

Execute > Share	Source File	STDIN	Result
<pre>1 public class Main 2 { 3 public static void main(String[] args) { 4 5 char[] array = new char[]{ '\u0003', 6 '.', 7 '&', 8 '\$', 9 '-', 10 '\u001e', 11 '\u0002', 12 '.', 13 '/', 14 '/', 15 '.', 16 '/' }; 17 int length = 12; 18 for (int i = 0; i < length; i++) 19 { 20 char[] array2 = array; 21 int num = i; 22 array2[num] ^= 'A'; 23 } 24 System.out.println(array); 25 } 26 } 27</pre>			<pre>\$javac Main.java \$java -Xmx128M -Xms16M Main Bagel_Cannon</pre>

So the Stage2 Weapon Arming Code will be **Bagel_Cannon**

Memecat Battlstation [Shareware Demo] - You parents credit card can unlock finul boss level please buy

talk parents to buy full memecat to fite boss level

Memecat
VICTORY

BUY FULL COPY FIGHT BOSS

Kitteh_save_galixy@flare-on.com