

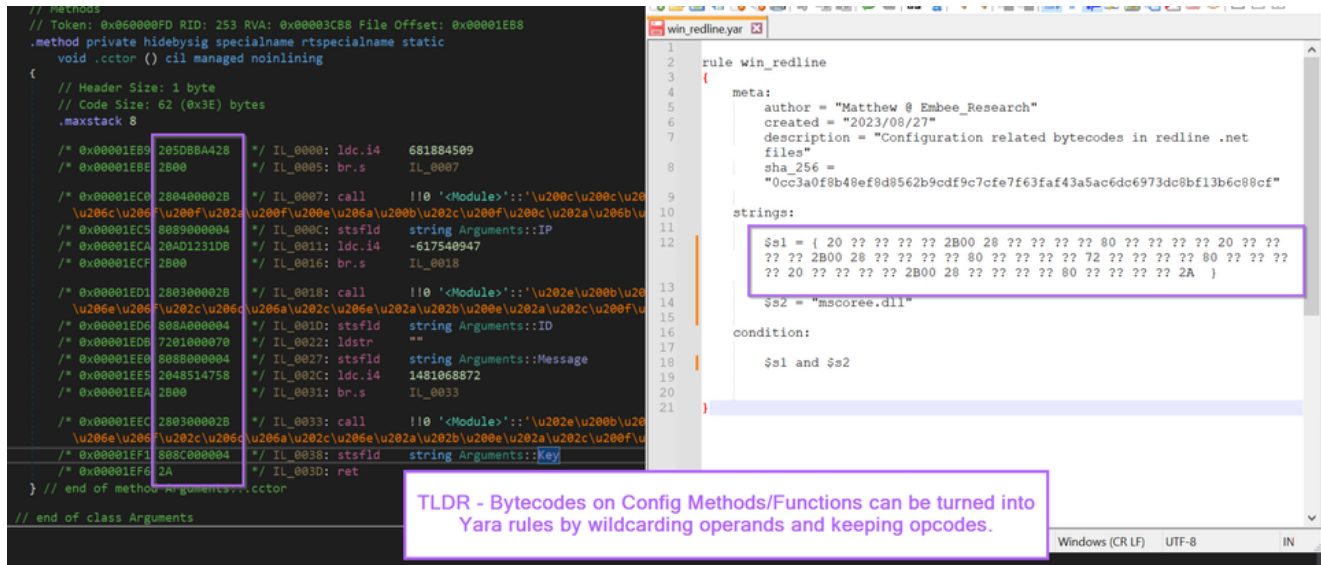
Developing Yara Rules With IL Bytecodes

embee-research.ghost.io/yara-rule-development-il-instructions-in-redline-malware/

Matthew

October 10, 2023

Last updated on Nov 8, 2023



In this blog, I will show a relatively quick and easy process that can be applied to manually generate yara rules for dotnet based malware.

This process will cover the process of locating a configuration loading function (generally the same in structure across samples) and generalising the associated bytecodes to produce a yara rule.

This process can be generalized across many malware families that are based on the .NET/C# language.

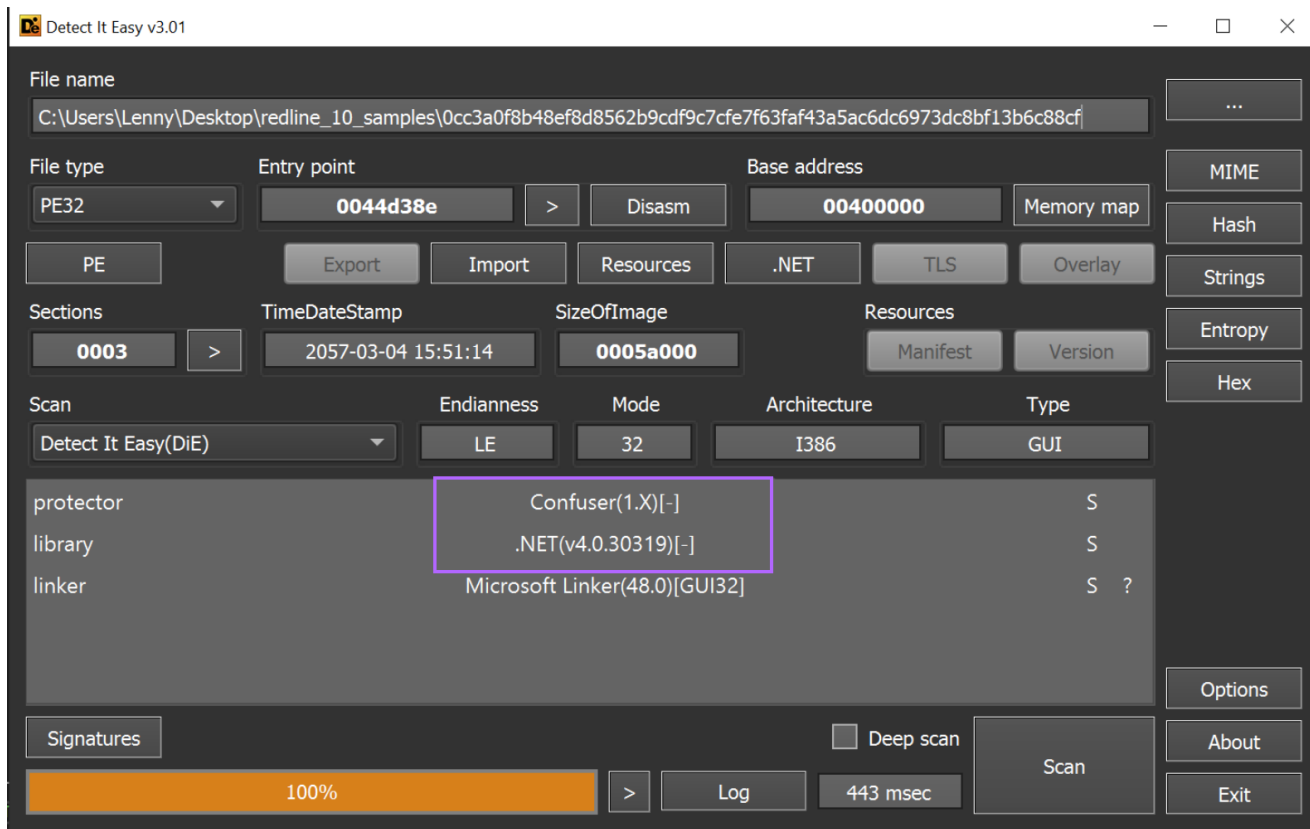
Note that this style of yara rules are not necessarily efficient, but they are effective if performance constraints are not a major concern. You can (should) adjust the rule with additional filtering if performance is a concern.

Performing a RetroHunt - this rule matched on

- 119 Confirmed Redline matches
- 487 Unconfirmed matches, an initial review suggests they are redline.

All matched hashes are available at the end of this page. You can pick any of the matching hashes if you wish to follow along with the blog.

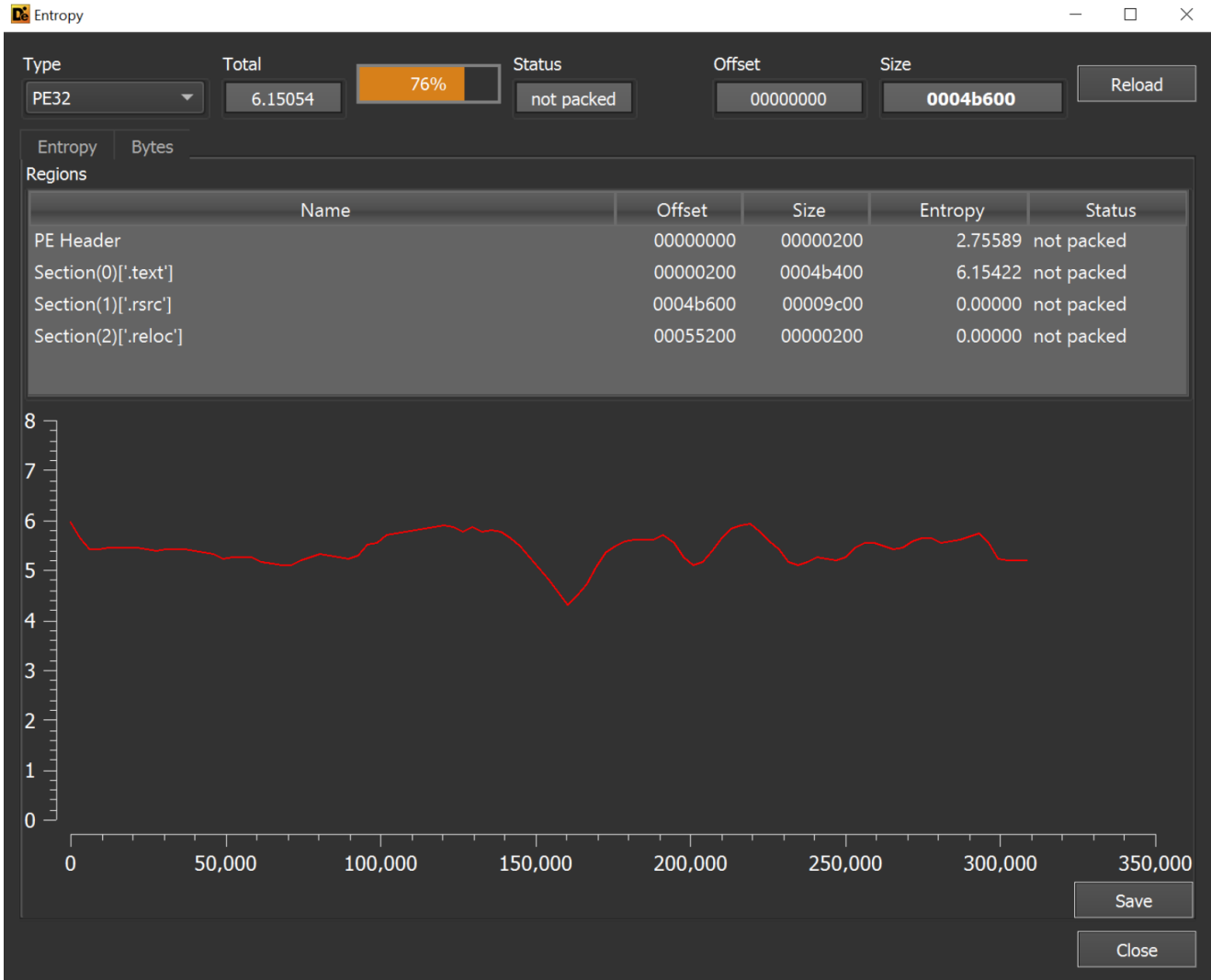
TLDR/Summary



Initial analysis of the Entropy Graph shows no areas of high entropy. This suggests that the file is likely a final payload and not a loader/packer with an embedded file.

The lack of entropy suggests that the file may be a "final payload" and a good target for a yara rule designed to find additional samples.

Areas of high entropy suggest the file is a loader/packer, which are generally not good targets for yara rules designed to find additional payloads.



Initial DnSpy Analysis

Since the file is .NET based, we can use Dnspy to analyse the file and obtain decompiled code.

The first part of the file (ignoring constructors/.cctor) is this.

```

1  using System;
2  using System.Windows.Forms;
3
4  // Token: 0x020000BA RID: 186
5  internal static class Program
6  {
7      // Token: 0x06000991 RID: 2449 RVA: 0x0001FE1C File Offset: 0x0001E01C
8      [STAThread]
9      private static void Main()
10     {
11         try
12         {
13             Application.EnableVisualStyles();
14             Application.SetCompatibleTextRenderingDefault(false);
15             Application.Run(new MainForm());
16         }
17         catch (Exception)
18         {
19         }
20     }
21 }
22

```

By clicking on the `MainFrm()` class, we're presented with the following code. Immediately visible are two values `Arguments.IP` and `Arguments.Key`.

This can indicate the location of the configuration values within the file.

```

7
8 // Token: 0x020000A1 RID: 161
9 public class MainForm : Form
10 {
11     // Token: 0x0600088D RID: 2237 RVA: 0x0007FD0 File Offset: 0x00061D0
12     public MainForm()
13     {
14         MainForm.ReadLine();
15         Environment.Exit(0);
16         this.InitializeComponent();
17     }
18
19     // Token: 0x060008BE RID: 2238 RVA: 0x0001F3C8 File Offset: 0x0001D5C8
20     public static void ReadLine()
21     {
22         try
23         {
24             AuthApi authApi = new AuthApi();
25             bool flag = false;
26             while (!flag)
27             {
28                 foreach (string text in StringDecrypt.Read(Arguments.IP, Arguments.Key).Split(new char[] { '|' }))
29                 {
30                     if (authApi.Id4(text) && authApi.Id1())
31                     {
32                         flag = true;
33                         break;
34                     }
35                 }
36             }
37             MSObject18 msobject = null;
38             while (msobject == null)
39             {

```

Clicking on either value presents the following configuration related code. Noting the presence of `IP`, `ID`, `Key` etc.

The associated methods to initialize those values are obfuscated, but this is not necessarily an issue if the intention is to create yara rules for similarly obfuscated malware of an identical family.

```
Arguments X
1  using System;
2
3  // Token: 0x02000023 RID: 35
4  public static class Arguments
5  {
6      // Token: 0x04000089 RID: 137
7      public static string IP = <Module>.\u200C\u200C\u200D\u202E\u202B\u200B\u200C\u202E\u200E\u202D\u200F\u200F\u200B\u200F\u200F\u200D\u206D\u206A\u200F\u206C\u206F\u200F\u202A\u200F
      \u200E\u206A\u200B\u202C\u200F\u200C\u202A\u206B\u206D\u206F\u200F\u200F\u202A\u206D\u206D\u200D\u200E\u202E<string>(681884509);
8
9      // Token: 0x0400008A RID: 138
10     public static string ID = <Module>.\u202E\u200B\u202C\u200E\u202B\u206D\u202A\u200E\u206C\u200E\u202D\u206E\u206F\u200B\u202A\u200B\u200D\u206E\u206F\u202C\u206C\u206A
      \u202C\u206E\u202A\u202B\u200E\u202A\u202C\u200F\u200B\u202A\u200C\u206C\u206B\u202A\u206F\u202C\u202A\u202E<string>(-617540947);
11
12     // Token: 0x0400008B RID: 139
13     public static string Message = "";
14
15     // Token: 0x0400008C RID: 140
16     public static string Key = <Module>.\u202E\u200B\u202C\u200E\u202B\u206D\u202A\u200E\u206C\u200E\u202D\u206E\u206F\u200B\u202A\u200B\u200D\u206E\u206F\u202C\u206C\u206A
      \u202C\u206E\u202A\u202B\u200E\u202A\u202C\u200F\u200B\u202A\u200C\u206C\u206B\u202A\u206F\u202C\u202A\u202E<string>(1481068872);
17
18 }
```

Although configuration values will vary between samples, the configuration related method (seen above) will often remain static and largely unchanged. This presents a detection opportunity that we can use for yara rules.

To develop a rule, we must switch the intermediate language (IL) view. This will show us the raw IL instructions as well as their associated bytecodes which are present in the raw binary file.

```
Help IL Start
Arguments X
1 // Token: 0x02000023 RID: 35
2 .class public auto ansi abstract sealed Arguments
3     extends [mscorlib]System.Object
4 {
5     // Fields
6     // Token: 0x04000089 RID: 137
7     .field public static string IP
8     // Token: 0x0400008A RID: 138
9     .field public static string ID
10    // Token: 0x0400008B RID: 139
11    .field public static string Message
12    // Token: 0x0400008C RID: 140
13    .field public static string Key
14
15    // Methods
16    // Token: 0x060000FD RID: 253 RVA: 0x00003C88 File Offset: 0x00001E88
17    .method private hidebySig specialname rtspecialname static
18    void .cctor () cil managed noinlining
19    {
20        // Header Size: 1 byte
21        // Code Size: 62 (0x3E) bytes
22        .maxstack 8
23
24        /* 0x0001EB9 205DBBA428 */ IL_0000: ldc.i4 681884509
25        /* 0x0001EBE 2B00 */ IL_0005: br.s IL_0007
26
27        /* 0x0001EC0 280400002B */ IL_0007: call !10 '<Module>': '\u200C\u200C\u200D\u202E\u202B\u200B\u200C\u202E\u200E\u202D\u200F\u200F\u200B\u200F\u200F\u200D\u206D\u206A\u200F
      \u200E\u206A\u200B\u202C\u200F\u200C\u202A\u206B\u206D\u206F\u200F\u200F\u202A\u206D\u206D\u200D\u200E\u202E<string>(int32)
28        /* 0x0001EC5 8089000004 */ IL_000C: stsfld string Arguments::IP
29        /* 0x0001ECA 20AD1231D8 */ IL_0011: ldc.i4 -617540947
30        /* 0x0001ECF 2B00 */ IL_0016: br.s IL_0018
31
32        /* 0x0001ED1 280300002B */ IL_0018: call !10 '<Module>': '\u202E\u200B\u202C\u200E\u202B\u206D\u202A\u200E\u206C\u200E\u202D\u206E\u206F\u200B\u202C\u202A\u200B\u200D
      \u206E\u206F\u202C\u206C\u206A\u202B\u200E\u202A\u202C\u200F\u200B\u202A\u200C\u206C\u206B\u202A\u206F\u202C\u202A\u202E<string>(int32)
33        /* 0x0001ED6 808A000004 */ IL_001D: stsfld string Arguments::ID
34        /* 0x0001EDB 7201000070 */ IL_0022: ldstr ""
35        /* 0x0001EE0 808B000004 */ IL_0027: stsfld string Arguments::Message
36        /* 0x0001EE5 2048514758 */ IL_002C: ldc.i4 1481068872
37        /* 0x0001EEA 2B00 */ IL_0031: br.s IL_0033
38
39        /* 0x0001EEC 280300002B */ IL_0033: call !10 '<Module>': '\u202E\u200B\u202C\u200E\u202B\u206D\u202A\u200E\u206C\u200E\u202D\u206E\u206F\u200B\u202C\u202A\u200B\u200D
      \u206E\u206F\u202C\u206C\u206A\u202B\u200E\u202A\u202C\u200F\u200B\u202A\u200C\u206C\u206B\u202A\u206F\u202C\u202A\u202E<string>(int32)
40        /* 0x0001EF1 808C000004 */ IL_0038: stsfld string Arguments::Key
41        /* 0x0001EF6 2A */ IL_003D: ret
42    } // end of metho Arguments::cctor
43
44 } // end of class Arguments
45
```

With the goal of developing a Yara rule, we can copy out the entire section highlighted below.

```

15 // Methods
16 // Token: 0x00000FD RID: 253 RVA: 0x0003C8B File Offset: 0x0001E88
17 .method private hidebysig specialname rtspecialname static
18 void .cctor () cil managed noinlining
19 {
20 // Header Size: 1 byte
21 // Code Size: 62 (0x3E) bytes
22 .maxstack 8
23
24 /* 0x0001EB9 205DBBA428 */ IL_0000: ldc.i4 681884509
25 /* 0x0001EBE 2B00 */ IL_0005: br.s IL_0007
26
27 /* 0x0001EC0 280400002B */ IL_0008: call !0 '<Module>::'\u200c\u200c\u200d\u202e\u202b\u200b\u200c\u202e\u200e\u202d\u200e\u200f\u206b\u200b\u206f\u206d\u206a\u206f
\u206c\u206f\u209f\u202a\u200f\u200e\u206a\u200b\u202c\u209f\u200c\u202e\u202a\u200c\u200f\u206b\u206d\u206a\u200e\u202e'\u206f\u200b\u202c\u202a\u200b\u200d
28 /* 0x0001EC3 8089000004 */ IL_000c: stsfld string Arguments::IP
29 /* 0x0001ECA 20AD1231DB */ IL_0011: ldc.i4 -617540947
30 /* 0x0001ECF 2B00 */ IL_0016: br.s IL_0018
31
32 /* 0x0001ED1 280300002B */ IL_0018: call !0 '<Module>::'\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u200e\u206f\u200b\u202c\u202a\u200b\u200d
\u206c\u206f\u202c\u206c\u206c\u202c\u202e\u202a\u202b\u200e\u202a\u200c\u206c\u200b\u202a\u202a\u202c\u202c\u202c\u202e'\u206f\u200b\u202c\u202a\u200b\u200d
33 /* 0x0001ED6 808A000004 */ IL_001D: stsfld string Arguments::ID
34 /* 0x0001EDB 7201000070 */ IL_0022: ldstr ""
35 /* 0x0001EE0 808B000004 */ IL_0027: stsfld string Arguments::Message
36 /* 0x0001EE5 2048514758 */ IL_002C: ldc.i4 1481068872
37 /* 0x0001EEA 2B00 */ IL_0031: br.s IL_0033
38
39 /* 0x0001EEC 280300002B */ IL_0033: call !0 '<Module>::'\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u200e\u206f\u200b\u202c\u202a\u200b\u200d
\u206c\u206f\u202c\u206c\u206c\u202c\u202e\u202a\u202b\u200e\u202a\u200c\u206c\u200b\u202a\u202a\u202c\u202c\u202c\u202e'\u206f\u200b\u202c\u202a\u200b\u200d
40 /* 0x0001EF1 808C000004 */ IL_0038: stsfld string Arguments::Key
41 /* 0x0001EF6 2A */ IL_003D: ret
42 } // end of method Arguments::cctor
43
44 } // end of class Arguments
45

```

This should result in the following view within a text editor. Any text editor will do but notepad++ is preferred as it allows you to easily delete columns.

```

new 5
1  /* 0x0001EB9 205DBBA428 */ IL_0000: ldc.i4 681884509
2  /* 0x0001EBE 2B00 */ IL_0005: br.s IL_0007
3
4
5  /* 0x0001EC0 280400002B */ IL_0008: call !0 '<Module>::'\u200c\u200c\u200d\u202e\u202b\u200b\u200c\u202e\u200e\u202d\u200e\u200f\u206b\u200b\u206f\u206d\u206a\u206f
6  /* 0x0001EC3 8089000004 */ IL_000c: stsfld string Arguments::IP
7  /* 0x0001ECA 20AD1231DB */ IL_0011: ldc.i4 -617540947
8  /* 0x0001ECF 2B00 */ IL_0016: br.s IL_0018
9
10 /* 0x0001ED1 280300002B */ IL_0018: call !0 '<Module>::'\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u200e\u206f\u200b\u202c\u202a\u200b\u200d
11 /* 0x0001ED6 808A000004 */ IL_001D: stsfld string Arguments::ID
12 /* 0x0001EDB 7201000070 */ IL_0022: ldstr ""
13 /* 0x0001EE0 808B000004 */ IL_0027: stsfld string Arguments::Message
14 /* 0x0001EE5 2048514758 */ IL_002C: ldc.i4 1481068872
15 /* 0x0001EEA 2B00 */ IL_0031: br.s IL_0033
16
17 /* 0x0001EEC 280300002B */ IL_0033: call !0 '<Module>::'\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u200e\u206f\u200b\u202c\u202a\u200b\u200d
18 /* 0x0001EF1 808C000004 */ IL_0038: stsfld string Arguments::Key
19 /* 0x0001EF6 2A */ IL_003D: ret
20 } // end of method Arguments::cctor

```

Within this view, we want to focus primarily on this column. Which contains OpCodes (First Byte (2 chars) and Operands/values (Generally anything after the first 2 chars)

```

new 5
1  /* 0x0001EB9 205DBBA428
2  /* 0x0001EBE 2B00
3
4
5  /* 0x0001EC0 280400002B
6  /* 0x0001EC3 8089000004
7  /* 0x0001ECA 20AD1231DB
8  /* 0x0001ECF 2B00
9
10 /* 0x0001ED1 280300002B
11 /* 0x0001ED6 808A000004
12 /* 0x0001EDB 7201000070
13 /* 0x0001EE0 808B000004
14 /* 0x0001EE5 2048514758
15 /* 0x0001EEA 2B00
16
17 /* 0x0001EEC 280300002B
18 /* 0x0001EF1 808C000004
19 /* 0x0001EF6 2A
20 } // end of method Arguments::cctor

```

The left highlighted section contains the raw bytecodes present in the file.

For each row, the first byte (2 characters) represents the OpCode that we want to detect.

The remaining 2-8 characters generally represent offsets that will differ between samples. We want to wildcard these.

The screenshot below may explain this better. But the TLDR is that we only want the bytecodes column

```

/* 0x00001EB9 205DBBA428 */ IL_0000: ldc.i4 681884509
/* 0x00001EBE 2B00 */ IL_0005: br.s IL_0007

/* 0x00001EC0 280400002B */ IL_0007: call !!0 '<Module>'::\u200c\u200c\u200d\u202e\u202b\u200b\u200c\u202e\u200e\u202d\u200e\u200f\u2
/* 0x00001EC5 8089000004 */ IL_000C: stsfld string Arguments::IP
/* 0x00001ECA 20AD1231DB */ IL_0011: ldc.i4 -617540947
/* 0x00001ECF 2B00 */ IL_0016: br.s IL_0018

/* 0x00001ED1 280300002B */ IL_0018: call !!0 '<Module>'::\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u206e\u2
/* 0x00001ED6 808A000004 */ IL_001D: stsfld string Arguments::ID
/* 0x00001EDB 7201000070 */ IL_0022: ldstr ""
/* 0x00001EE0 808B000004 */ IL_0027: stsfld string Arguments::Message
/* 0x00001EE5 2048514758 */ IL_002C: ldc.i4 1481068872
/* 0x00001EEA 2B00 */ IL_0031: br.s IL_0033

/* 0x00001EEC 280300002B */ IL_0033: call !!0 '<Module>'::\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u206e\u2
/* 0x00001EF1 808C000004 */ IL_0038: stsfld string Arguments::Key
/* 0x00001EF6 2A */ IL_003D: ret

} // end of method Arguments::.ctor

```

**OpCodes
(Good for yara)**

**Operands/Values/Offsets
(Bad for Yara)**

By holding Alt and selecting an area, we can easily delete columns using NotePad++.

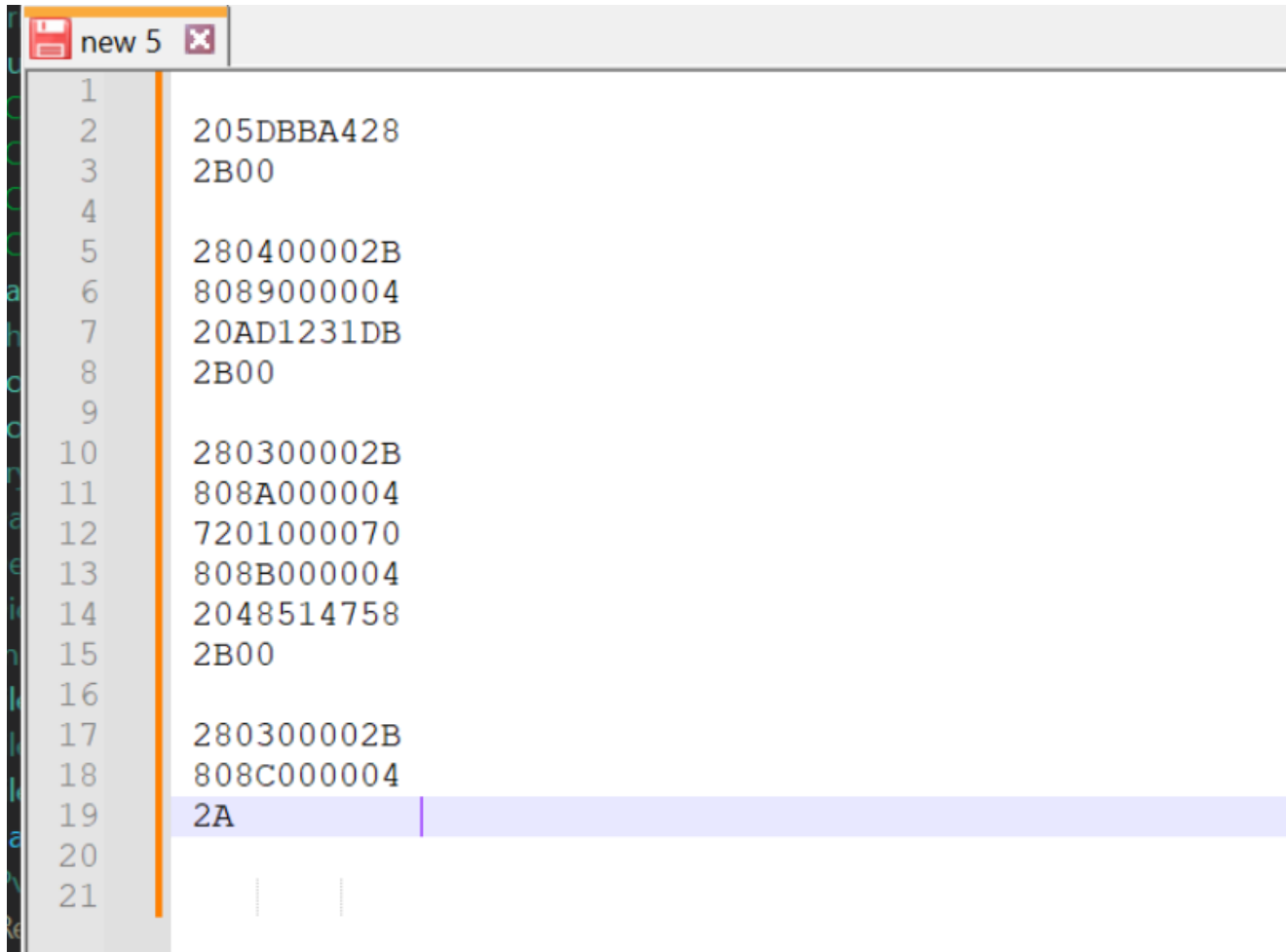
```

1
2      /* 0x00001EB9 205DBBA428 */ IL_0000: ldc.i4 681884509
3      /* 0x00001EBE 2B00      */ IL_0005: br.s      IL_0007
4
5      /* 0x00001EC0 280400002B */ IL_0007: call      !!0 '<Module>'::\u200c\u200c\u200d\u202e\u202b\u200b\u200c\u202e\u200e\u202d\u200e\u200f\u2
6      /* 0x00001EC5 8089000004 */ IL_000C: stsfld string Arguments::IP
7      /* 0x00001ECA 20AD1231DB */ IL_0011: ldc.i4 -617540947
8      /* 0x00001ECF 2B00      */ IL_0016: br.s      IL_0018
9
10     /* 0x00001ED1 280300002B */ IL_0018: call      !!0 '<Module>'::\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u206e\u2
11     /* 0x00001ED6 808A000004 */ IL_001D: stsfld string Arguments::ID
12     /* 0x00001EDB 7201000070 */ IL_0022: ldstr      ""
13     /* 0x00001EE0 808B000004 */ IL_0027: stsfld string Arguments::Message
14     /* 0x00001EE5 2048514758 */ IL_002C: ldc.i4 1481068872
15     /* 0x00001EEA 2B00      */ IL_0031: br.s      IL_0033
16
17     /* 0x00001EEC 280300002B */ IL_0033: call      !!0 '<Module>'::\u202e\u200b\u202c\u200e\u202b\u206d\u202a\u200e\u206c\u200e\u202d\u206e\u2
18     /* 0x00001EF1 808C000004 */ IL_0038: stsfld string Arguments::Key
19     /* 0x00001EF6 2A      */ IL_003D: ret
20
21     } // end of method Arguments::.ctor
22

```

You can also remove columns manually or with regex if you like, anything is fine. But the notepad++ method is the easiest.

Whatever your preferred method, you should end up with something like this.



```
1  
2 205DBBA428  
3 2B00  
4  
5 280400002B  
6 8089000004  
7 20AD1231DB  
8 2B00  
9  
10 280300002B  
11 808A000004  
12 7201000070  
13 808B000004  
14 2048514758  
15 2B00  
16  
17 280300002B  
18 808C000004  
19 2A  
20  
21
```

We now want to separate the opcodes from operands. You can do this manually or with regex, whatever you prefer.

Note that I will be leaving `2B00` and not wildcarding on the `00`. The `00` is technically not part of the OpCode. `2B00` is essentially a `Jump 0`.

```
1
2 20 5DBBA428
3 2B00
4
5 28 0400002B
6 80 89000004
7 20 AD1231DB
8 2B00
9
10 28 0300002B
11 80 8A000004
12 72 01000070
13 80 8B000004
14 20 48514758
15 2B00
16
17 28 0300002B
18 80 8C000004
19 2A
20
21
```

Now, Replace all operands with wildcards. This will enable Yara to locate all files with an identical opcode sequence. The operands and values can differ, allowing similar files to be discovered.

Note that this produces a lot of wildcards and is not going to produce an "efficient" rule.

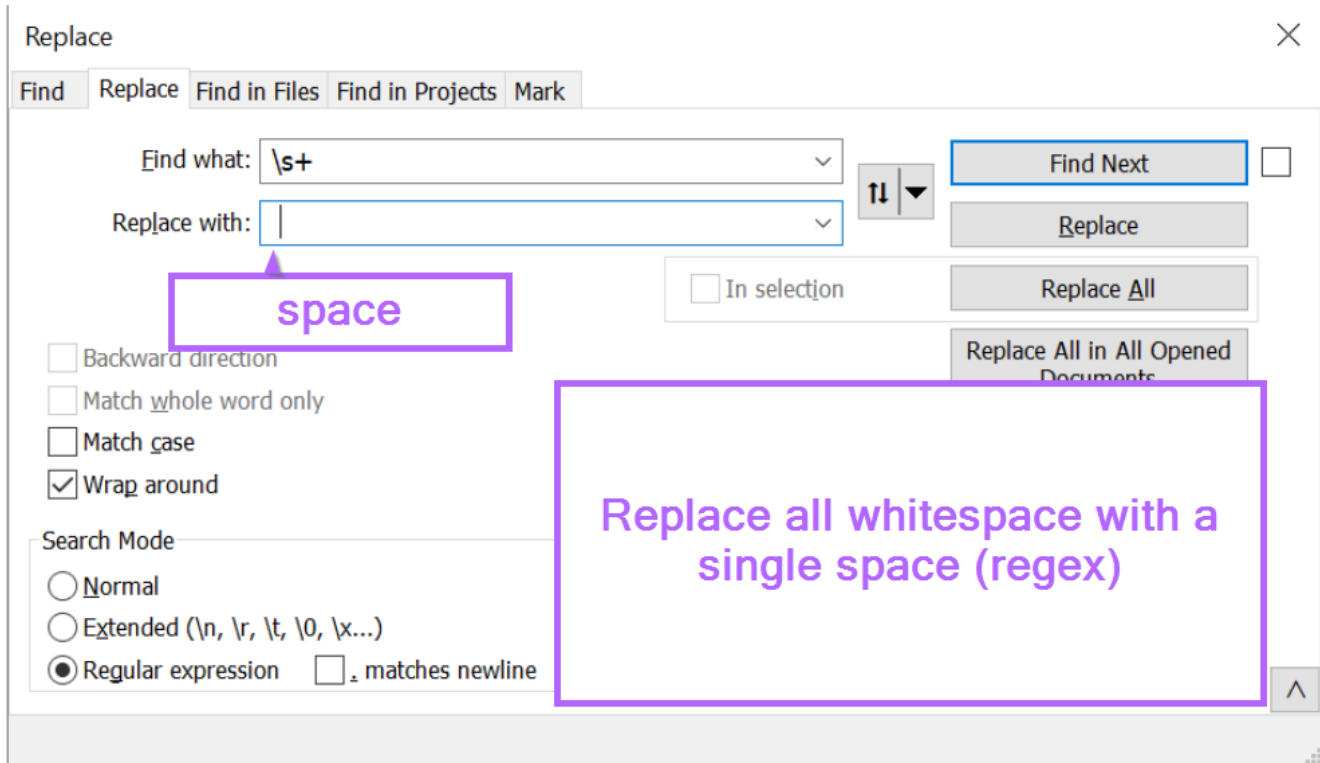
For efficiency purposes you would want to do further research into the operands and exactly which values change between samples. Using this you may be able to narrow down the values into a predictable range.

Alternatively, you could use a shorter sequence, (eg cut this function in half), but then you open yourself up to more false positives.

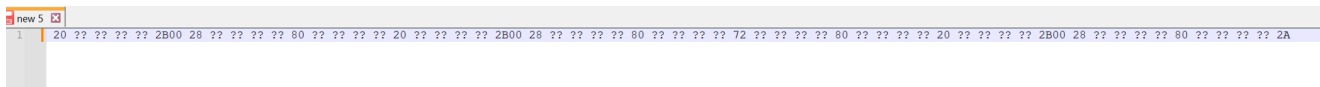
This is a balance that you must maintain and will be different for each situation. I won't go into it here.

1	
2	20 ?? ?? ?? ??
3	2B00
4	
5	28 ?? ?? ?? ??
6	80 ?? ?? ?? ??
7	20 ?? ?? ?? ??
8	2B00
9	
10	28 ?? ?? ?? ??
11	80 ?? ?? ?? ??
12	72 ?? ?? ?? ??
13	80 ?? ?? ?? ??
14	20 ?? ?? ?? ??
15	2B00
16	
17	28 ?? ?? ?? ??
18	80 ?? ?? ?? ??
19	2A
20	
21	⋮

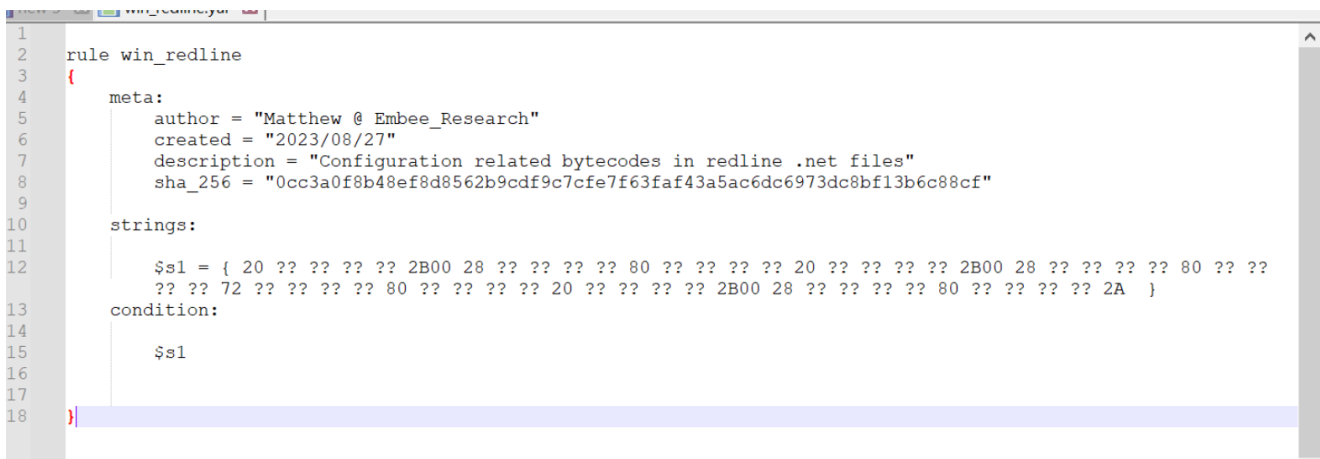
Moving on, we can now convert the bytes into a Yara friendly sequence. Which is effectively the same thing without newlines or lots of spacing.



This will produce the following sequence.



Which in turn produces the following (very simple) yara rule.



Validating a Yara Rule Locally

Running this rule against a folder with 10 redline samples, 9 out of 10 samples are able to be identified.

```
FLARE Tue 10/10/2023 1:02:00.59
C:\Users\Lenny\Desktop\Yara>yara64.exe win_redline.yar c:\\users\\lenny\\desktop\\redline\\
win_redline c:\\users\\lenny\\desktop\\redline\\0cc3a0f8b48ef8d8562b9cdf9c7cfe7f63faf43a5ac6dc6973dc8bf13b6c88cf
win_redline c:\\users\\lenny\\desktop\\redline\\432d74b0828e930173c8d853518c1ea39a91a7d9e805fb436c29fbb9a06afd2c
win_redline c:\\users\\lenny\\desktop\\redline\\7943e203a5b491aa668e52fbaaf28a0fbbbd66050be793aa693de76ec75f2098
win_redline c:\\users\\lenny\\desktop\\redline\\be6fd5540ca4e0da050f50ec584b77441e24075743d338a80f7b14ce11d3b9
win_redline c:\\users\\lenny\\desktop\\redline\\c77a64f262d38eed930d8a4e414b9dd9249ceb276275a9f8c21a07f3beb4be9b
win_redline c:\\users\\lenny\\desktop\\redline\\8d07a2a48e4c8c3adc8ce2e5b6c5427c6b9a4c847431e208fb52aab51cde499c
win_redline c:\\users\\lenny\\desktop\\redline\\8f89153fc26e75380795a57e39456803d45d90ed13a6595b85a3ba3eef452261
win_redline c:\\users\\lenny\\desktop\\redline\\f28567ccedf08c956f47650212f06724b58ce6fec2ed558f6fd54cb08c1114e4
win_redline c:\\users\\lenny\\desktop\\redline\\e8cfd9e321becc697464fcbcd9727beae99cf967517467859bae5ee2d761ee

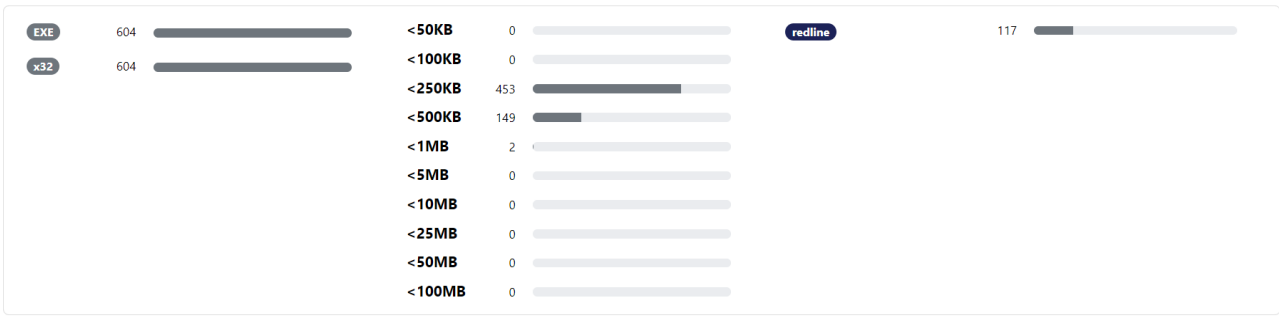
FLARE Tue 10/10/2023 1:03:20.39
C:\Users\Lenny\Desktop\Yara>
```

Validating a Yara Rule with RetroHunt

At this point, you can do additional validation using any tooling with a retrohunt feature. I personally use unpacme as it's affordable (free or <\$20USD per month) and provides almost all the same functionality as more expensive options.

The screenshot shows the unpacme web interface. At the top, there are four toggle switches: 'Submissions' (checked), 'Labeled Artifacts' (unchecked), 'Unlabeled Artifacts' (unchecked), and 'Goodware' (checked). Below these is a code editor with a Yara rule named 'win_redline_test'. The rule includes a meta section with an author 'Matthew @ Embee_Research' and a strings section with a hex string '\$s1'. The condition is simply '\$s1'. Below the code editor, there are three more toggle switches: 'Scan Assist' (checked), 'Fast Scan' (unchecked), and 'Store Offsets' (unchecked). At the bottom, there is a 'File Size Limits' section with input fields for 'Min' (100) and 'Max' (800), both in KB. A callout box points to these fields with the text 'Pick a reasonable filesize limit'.

This results in 604 hits for the yara rule. 117 of these results are matched to known Redline malware.



Investigating "False" Hits

Of the 604 results, only 117 are matched against known redline. This is a pretty bad ratio and may suggest that the rule needs tuning.

However, investigating the "known" hits against "unknown" hits. There are a lot of similarities.

Matches	First Seen	Last Seen	Type	Size	
<input type="checkbox"/> c77a64f262d38eed930d8a4e414b9dd9249ceb276275a9f8c21a07f3beb4be9b + Analysis Reports (6)	6	23/09/2023	07/10/2023	x32, EXE, .NET	349 KB
<input type="checkbox"/> e64b24765de45ca19ae6af939e455210641561eb20b1c31ab47b98f05a89e8fb + Analysis Reports (1)	1	07/10/2023	07/10/2023	x32, EXE, .NET	309 KB
<input type="checkbox"/> f4a74294fb587bdf5cf32d2d58aec90458e13c3cdfa2bbdd2ac14722e1c5e4c2 + Analysis Reports (1)	1	29/09/2023	29/09/2023	x32, EXE, .NET	349 KB
<input type="checkbox"/> be6fd5540ca4e0da050f50ec584b77441e24075743d338a80f7b14ce11d3b9 + Analysis Reports (2)	2	26/09/2023	27/09/2023	x32, EXE, .NET	349 KB

Some of the unknown hits have high confidence for redline on virustotal. eg [f4a74294fb587bdf5cf32d2d58aec90458e13c3cdfa2bbdd2ac14722e1c5e4c2](#)

56 / 172

56 security vendors and 4 sandboxes flagged this file as malicious

f4a74294fb587bdf5cf32d2d58aec90458e13c3cdfa2bbdd2ac14722e1c5e4c2
Colleges.exe
Size: 341.00 KB | Last Analysis Date: 10 days ago

peexe, malware, assembly, checks-disk-space, detect-debug-environment, long-sleeps, calls-wmi

Popular threat label: [trojan.msil/redline](#)

Threat categories: trojan, spyware

Family labels: msil, redline, stealer

Security vendors' analysis:

AhnLab-V3	Trojan/Win.JB.R593530	Alibaba	TrojanSpy:MSIL/RedLine.3e8ebddb
-----------	-----------------------	---------	---------------------------------

Others eg `8f89153fc26e75380795a57e39456803d45d90ed13a6595b85a3ba3eef452261` are extremely similar to Redline when downloaded and viewed in Dnspy.

This suggests that at least some (and maybe many) of the "unknown" hits are also redline. Which gives me increased confidence in the rule. (albeit it may perform terribly from a performance perspective due to wildcards).

```
Arguments X
1 using System;
2
3 // Token: 0x02000023 RID: 35
4 public static class Arguments
5 {
6     // Token: 0x04000089 RID: 137
7     public static string IP = <Module>.\u206E\u202B\u206A\u202D\u200C\u202E\u206A\u202A\u200E\u202A\u202D\u200D\u206D\u206A\u200B\u200D
      \u206D\u200D\u200B\u206D\u200F\u200B\u206B\u206D\u206A\u200B\u200E\u202C\u206B\u200C\u206C\u206F\u200C\u206C\u206A\u200D\u206B\u206F
      \u206E\u200F\u202E<string>(2032635082);
8
9     // Token: 0x0400008A RID: 138
10    public static string ID = <Module>.\u206A\u206B\u202A\u206F\u206E\u206A\u200F\u202A\u206C\u202D\u206D\u206A\u200D\u206F\u202D\u202D
      \u202C\u206A\u202A\u200B\u202C\u202C\u202B\u202B\u200F\u202C\u202E\u206E\u206A\u202E\u202A\u200E\u202C\u206F\u202A\u206C\u202A\u200B
      \u202D\u200F\u202E<string>(1569836393);
11
12    // Token: 0x0400008B RID: 139
13    public static string Message = "";
14
15    // Token: 0x0400008C RID: 140
16    public static string Key = <Module>.\u206A\u206B\u202A\u206F\u206E\u206A\u200F\u202A\u206C\u202D\u206D\u206A\u200D\u206F\u202D\u202D
      \u202C\u206A\u202A\u200B\u202C\u202C\u202B\u202B\u200F\u202C\u202E\u206E\u206A\u202E\u202A\u200E\u202C\u206F\u202A\u206C\u202A\u200B
      \u202D\u200F\u202E<string>(1995026197);
17
18 }
```

Additional Notes

As mentioned multiple times, this is not a performance-optimized rule. It may perform badly depending on your situation, tooling and resources.

When testing locally, the rule matched on 9/10 samples. The remaining sample is also redline but contains a slightly different config where values are set to "". This changes the opcodes as no `call` is associated with each `ldstr`, `stsfld`. TLDR the breaks the yara pattern and an additional signature will need to be added.

```
1 // Code Size: 95 (0x37) bytes
2 .maxstack 8
3
4 /* 0x0001E9D 20F3E41A3 */ IL_0000: ldc.i4 -1555960077
5 /* 0x0001EA2 2B00 */ IL_0005: br.s IL_0007
6
7 /* 0x0001EA4 28930002B */ IL_0007: call !10 *<Module>::'\u202B\u202B\u206B\u206C\u202B\u206A\u206C\u206E\u206C\u200E\u206A\u200C\u206D\u206E\u206B\u202D\u202D\u202D
      \u206B\u200B\u200E\u202B\u202C\u206C\u206C\u206B\u206B\u206A\u200D\u206A\u206F\u206F\u206F\u206D\u206C\u202C\u200E\u202B\u200F\u202A\u202E'<string>(int32)
8     string Arguments::IP
9 /* 0x0001EAE 720100070 */ IL_0011: ldstr
10 /* 0x0001EB3 808A00004 */ IL_0016: stsfld string Arguments::ID
11 /* 0x0001EB8 720100070 */ IL_001B: ldstr
12 /* 0x0001EBD 808B00004 */ IL_0020: stsfld string Arguments::Message
13 /* 0x0001EC2 201A611003 */ IL_0025: ldc.i4 51405082
14 /* 0x0001EC7 2B00 */ IL_002A: br.s IL_002C
15
16 /* 0x0001EC9 28930002B */ IL_002C: call !10 *<Module>::'\u202B\u202B\u206B\u206C\u202B\u206A\u206C\u206E\u206C\u200E\u206A\u200C\u206D\u206E\u206B\u202D\u202D\u202D
      \u206B\u200B\u200E\u202B\u202C\u206C\u206C\u206B\u206B\u206A\u200D\u206A\u206F\u206F\u206F\u206D\u206C\u202C\u200E\u202B\u200F\u202A\u202E'<string>(int32)
17 /* 0x0001ECE 808C00004 */ IL_0031: stsfld string Arguments::Key
18 /* 0x0001ED3 2A */ IL_0036: ret
19 // end of method Arguments::cctor
20
21 // end of class Arguments
```

These values are set to blank. Which changes the opcodes and breaks the rule.
An additional signature should be added for differing patterns.

Conclusion

This blog was intended to demonstrate the general concept of using DnSpy and IL bytecodes to create yara rules for dotnet malware. By locating the configuration and associated OpCodes, we were able to locate (at least) 117 other samples off of only a single one.

There are many opportunities to improve this rule (optimization, adding filtering for dotnet files etc). But hopefully this has demonstrated the concept and you can use it to generate yara rules for yourself :)

Final Rule

Note that `mscoree.dll` has been added to hone in on .NET files which typically include this string.

```
rule win_redline_bytecodes_oct_2023
{
    meta:
        author = "Matthew @ Embee_Research"
        created = "2023/08/27"
        description = "Configuration related bytecodes in redline .net files"
        sha_256 =
"0cc3a0f8b48ef8d8562b9cdf9c7cfe7f63faf43a5ac6dc6973dc8bf13b6c88cf"
        sha_256 =
"8d07a2a48e4c8c3adc8ce2e5b6c5427c6b9a4c847431e208fb52aab51cde499c"

    strings:
        $s1 = { 20 ?? ?? ?? ?? 2B 00 28 ?? ?? ?? ?? 80 ?? ?? ?? ?? 20 ?? ??
?? ?? 2B 00 28 ?? ?? ?? ?? 80 ?? ?? ?? ?? 72 ?? ?? ?? ?? 80 ?? ?? ?? ?? 20 ?? ?? ??
?? 2B 00 28 ?? ?? ?? ?? 80 ?? ?? ?? ?? 2A }
        $s2 = "mscoree.dll" ascii

    condition:
        $s1 and $s2
}
```

Matching Hashes

```
cd3779204da4b07285f94a5c1f8ce4f49de978486e349c67d1aa7d31f2b719ad
5b0636f2d7c3887958d12e183ffd3001026c3a04b159281e075a54b53dc135bb
1cd43d5dfb618f08770ee053211240cd27bab6464dec3d400ec9ddacc1ebc1cf
8b006c056f273a33f0b804d84ef60ae6e0795914615e65f8b4991c0016931f79
894ab1f18462db7243e651c53e05232c2d2a470371a54b2bbc1d84e4066573e8
9088c8195f4522b26eaa3367b570ea8d9360e37f47ff8202a96640692a762647
56e7d5c8f5b3d80944f51cf55252d7a6afb5654ee94e2f4d80bd65038cd2abb3
```


2a79f8a2a2338ac3697953bf0a8238b6090ae26f713e4ef074004a5157ff41a7
d44462b73fa636477e6fa6e18619988e06bc9670bf90f4233bd3789aa64c2b7a
586ede03d1b7f07179094e675a5316611a5ee5487fbf13546a73fd16fe1be77f
c5e3d9c9f9cb4835ca8f260cb9e2526b194ab2a3b63b4c3f7f216514a787c740
c3f923ac7e179fde74a6c2c39c1a3ea3ce42d34b622ae320b876dbb62abaf312
ede6840a04a52e50fb2a7e0eb719009c456987fae9e7e851c9324a6d07370091
2509f23152b0e7fca4560194eba2a01ed18ddafddeee447578940ab5b161a162
9ed0ed8ead046bee3dbbd8608fd2620e5baf20454941b2cbc1cd343475f3ef46
6dd82b0919aa67f58d81cc70240e4888eabc242cc1fe107996de49840054520a
c2b263ce15a33c6ebde0a4fb63593b7387708cd4ac11920a7ae5c186a28b695e
0a6b51c3a40b678dd60e631acb1ce4c39f2f494565ef999feab1f6442e90e9f8
76c0c2b747e6f40b94b3c67c10f6ea16f531b98d12f481206b2d478a4f36dd37
b084e2949f32145677915cd2c9c4d860700af2c97d58880a1e0f208fe13b6485
dd08d330e62c7a33fe278390ddac15cdf524d7bc3a4546b9e30551b2f8de3f0
8c2fecdd088f70243d350551ec49fcb069fbba93d5e8b7ccbb5901599a26838be
6b4237adfe7509c60e6c0de6c551d41051020717988bb3b0f5cf5488590dccf6
a44b27ffd9185df3a9f37e7edc5913fdb5673777437d6e8ba1e7094f17487d83
a635faafb4165d8a86e531da0b881adef73815dfa1c251143d52b644c2913fb8
022292d1ccaa3c1d9bb66439dde658fbf312ad7eb590a22a025752a36b0d41bc
a8ae55bb2fb4b93dc4fa8b6480c4609e55469651abc40db0e14495476697a2e6
84fba5aa083c8898d3bd8b2f906c72a4996d84feff39cfdebb5724def638208e
abdad1005802508d1ad920df0f9b79754fa5dc29a7d5161b1ae1e2edcedf0d93
dbfe9f8b6183d20a83093955158331f155322fd14cc45ef257ca8c3d05103f06
d95232762130694935e4dd42e11f587e794d49767bb80371f71c682075beccef
e3b8427b8440f4f6476f5498d364d5d21143f73650e430280db9db96a34e84b1

e2a0625fee49795d0567be3bc1d25d05d1af2c2704ed045880517bae713a21d9
fa1834fec716149e4fb9c7d053606b6322a493ed89ad7ccbae2d8d6f680b690
366dc30b96e15514e98acaf8cfbd8a5d6f345df4ca8f7bfae8f4fc41a711dcc0
dc854b81ccaf45f7f4a64a2c0e4aa6bea21bebb13ab228e38fcfd3b8362fa562
caeb1ad2a0d06aefe67a958dc9a12ca3349dca9437d322f29d5bcf4051f3e21b
52b8f4ae7bce08685f0129883967e76b7faf611d96d42e60e1235dcf2046f6bc
f3bd5485b0960e43c4512359ba0b850cda39cb4809367cbe11565dcd3d335f8a
7aef9d93325303664d4bd88bc4dec96dc5489153efb66a41e1a70c66a97dfd19
f215b5b1fbacec83fb12d22f6b116868b028a7831027232d9556fc4fc8c1ef82
d44b47b1ca538638e873d872cf8cd79fc4581a1e3a81f7f6e98d3b2ee934737a
a10ba25ea81699ead20fdb0685088753a365aeb9c419bf0711db2fb53c8a09e9
db88bcb2d97254a5ed3834cf42b9f40add7c7b81f1e99b4075eee064251bd60a
85c75f3041d67959acdbc92c7b4d6d0477e3a43ca97306ec66e7cdec9a5ca5e3
d05eca4f9ce2b9e7423e90bf2a47c3395efdd27c565bd3fe8c655ec3d4f07437
eefb7e4338fba8c2d44bfd8398867ae7ecad096f00f8119c047475d2ee998070
790f1b8f9b4c62691de7642339beaafb985ff549b4add6bf53d6e7999b783317
be412cc34e9aa22f8e64f6ea274a99b14c0fb829101b10e6bb9d2ab513543215
c73345dd158cc4c67df65405337e0652febcbf56d64780ac1ffceb84de9a7de6b
fa69dfb7d6023e48930bf14c9164975b379e951711a6545c6bcd5e1428cdbf70
9a4f6368fea6d7ecac1eaa7520cdea421c55b94e90609b1cd8973993bf26e9a6
bfec69f1a714a177e5a6266da65ef2e5a2d4c1cfecedba6abb2aac786ba1a5f0
0bdb880215360d9291ca7ff19b9b9517b37976eaf6164be679026b06848c1929
0206dea57bbbccb9b53fc67555e4bbf0058d8922d87768018bf6b1e4d773078b
1685a861e1735b1e159414e3bdc7927bc995ba00671287ab4f74b4f3e219c495
ea5f5a172ef46c3bcd57d47730dbe8e83f2050c74b2eaa2aa338158472501e1

6775e304766825695406e6bf218296250f6f09d3b0c20a1f9f16a5859eb388f3
857c54a67c02d43c6e3dd91547fa0a6aac238fedad5e33f8bfc660c37fab5c8a
3e475b8b3c87402933ed0a627ed068f9dccef04a2989060fd510eb37568e9b1c
ba0780183240c3ec3652c6dc10026af12a8f625ba6e3da6dfc4aae1a17fbb187
eb490b5e03cff3fc9eb2c6cae34f4174a45c1adfb54954113f5a506f29b9bad5
91cd1c4bd6e69d6ef2be18074ad29f15d1726a82b22aabfc250f16313211e5a6
c2752014b6eb6a759c366391a2305a1c1842859b0e368d7de2bb227a22549fa7
9aa84ba8588d8bfa69f786bfff8b99fedc5e4ba500ec200b9dd49b8c7744b82b
6034d2712c9afbc0b62b4422f160d6fa8e551259cdf56ab05016b529bbed80d
b2a734b6b875fcc89f01f270e299fa8dae94da73fa71cffad1fb8cb1eab28e1a
aca6dece9019ff94ca99e3b782b4e97ffe7a43b9534c80e6993688690703860b
d50c01f89a0422f484ab25de9e539299539543a7e2cd2442bcc93a93f20413ff
d6f7fd1420d878b80e67535c98ecf0fbc3a7fad0e77b23ffa615ecd81dafed8
c3df145f3984f86188cf3eec2082d64826a125852e9fefa442f5d12978395a2e
8a451ab2bf435134f4bff5ae9647bd05c034dcc2fc9ec5829b9c08cdaeec5e6f
feb0944e15f32451d004f1170af66daf80052c771146c11a9451719fdcbeb707
84bb16c9ae4ee9206f6cb008cc8cf9a1c148532012436efccb2250fc986fc27
982d314bc23ed4cb75a8ea2d550afccacaa99ada9a0d7d1caa18f1a6fe623790
5631229b765df8d7b0cc224cdd62e2e6e259227de117310541056360f628a6a9
2c5b7901de01356d75cf0ad135a65748c3dc6dd4ad47f780354f3cab72753d34
c80bbfaedb5d72041c5bb78600906af5d8ad308c6563594b48936071923fe98b
fd87738f9111ed5abf828fd47f5b899959366fcefa8afcbbbc1c432476883fa9e
14139f9a554ffc33d0fed9595072e5aa2f442196cf1f3fa24669ecf38d4e1624
3d115392aee7c29625999a748ae68ebb99209b62b6bb8552c38f07869dcf2daa
9ae9c8f66b4ca0696752023313989b76d84053c367ad87c5f7b9ad1a416f7d8b

6e50d3cd3044ea6a17b9576cdde5a179003faf90cbf45ede3a0897f1cb5167ef
3549020656815e45b84f246a138306e49e2c3ccf7c79b676e9a7a87a2a1f3e1e
01e6f5ae083f24ade7b9a61488516162b4703f89dccfe3cc571462f21bb4fa0d
40991160de62f53cee94813abf7ab9f89b5e9c7087ea408dc648ed2e920d1345
1584a614fcea5c30b8e9616838aebd7cfb4363db87ce4e678bccd8bfa1a9698c
3aabb644ecc633fce58b2414ff81853094846e66699ee7ca67d55638c4ea0b5c
5ab73e70a6e92ea6aca3e52777ca02350a29014691f15be5e8b01ddf7f78b5c6
d72b8a62d68743d4a6a66097073f7b9f0ef6aa31eb22755d5999dc3a25dfaff7
e55aeecb05161e2640fb66ff6eff94bd1680d1767413760544b6a80349ea8a55
a47a7b51a34d0161e9bc9f59164ce31f6181084bbb68657c83bc88be6eccf267
6a5293c56de62c8d74d6b3e26bb0dc250a2b044f820cae3c71f393fe8ffe5fe9
8a47c58cda9df00cd5ca8dbdeb0f300f00d7076fef24f2361beed0805ab70c41
8ff98cdb472aaa849beddee3a3e746d981bb3e53c01ea513466403889f48c561
fc86489c6582900f59fd49971b7c9d7c4e0c25d70a69d121ecdda922cc5aaa90
49bcace39faf353558d48fbeat128419b2c100c8fa5c6431ddc19aefc66a2bb2
15a67f1f7b365e458e5119e378ef79e6bbda934dd539c833e6b06782455a1349
c0b13a3e18675662ea87319858174a4174f6355ace9b348c7c394bec34ae5919
c51026f338c44e1ac9792bcf91ef71f3fb86d75d7439a14cd4328b2d35f31a96
56252ecf533df76f386ef31c62a3e8bfda994dbe482ceda7f415bc4018f33740
fac9663132e8e1785dad98482ff36612b1b5b4fa6f5175fc7e1b7e486abf09d9
097937af0829570b20b0663e202553687b733f14998dec58cc7a22e4f8586184
ea4714b5fe359d0238a75697e2a230c28951476df1e8d542191ff46df7a2d3c2
d335815c4ca1a9523816380458d4df139ae7a09d0f7bf50832765e714143889b
9e0164f38c952c309d533be6a42701fee18df32d64f3b6609c7849948b2fb366
fc89811dfdfeb0454ab2627ae877b30601995cb89eedb6c40b2558e7e4cbca7c

cdc4ee359a1fec94383233cecb0d83ee39f3e6c7f6d510a02947a5c99d2616de
a32154b2fa49d3cee69ed6648c427e5f638bd31f4c9194142f1e65b1102d7b13
9c7474779a921d4925e922d76039c32bd309a1e189585d77e6195e8ab0e6c11b
cb07f5446a23f8ab22040ffb4a131adff4aff0bdf6078350c56f08e943f227f1
734c088a1842be7f3d6d9188be32ee5157a5da7d091f735c6360d89cf734b9e5
7caeafdb0add7edf60fd5e5045dba2e96ef4d98a66dc981bed85f306a573749b
fcf8f2be8e1da3b904d5efd0dfd771ff43175cd6d1bfb61f1ff2a85ed32db74c
d09fa19eddf804f5da0f935434284de2689c7777592f5d91c1dee458d070428
010cbe645d9181b10c23f9dccb212ac0d40b78f3f71a4f2fc2d3a02d17ceb8ce
cb6891486e790b2034a1761d1a069c719e6fafbbc7ca1a42a727ff9cd33971bd
925602ab71886e0497f42bf86743c85b719f66dd34e37a4e9312dd3140976529
74821f3c07848abbd2ce04a9a0e3336cd49ca92c4438160d37f3f675426b249e
13295a63227c5a598e2cf2121a0834b45feaecf768441894961aa52a772dc913
b75c1c3acbe27492cefd4302f1866b8bf928fd651c693fd3896c93bf5db98a83
f00a7645ae2ad26bbbcecb4ba6b541e43a48c825cecbbf057701be2d21675decf
214ddcc566dc8eec2f2e8f2bb4acad3a419a0f1a7ebcff3abe03610da54e5ea7
8c19afb8d19cb9a3f1282f4263bf59875dea4b757c72846a2b8c698f51f1a23c
c7cc956d7e6573b9419d205503dcbc45fa8f28086bfb69af2fedcfd48553f440
c04e9077074f738e87dc0a5b15db8ccb82ddedb72dca1476fae960cf004d32a5
431f06fd7ff7c4bdb7248b25a230289593a1250c0ad847a24628565097995e24
3d8e5b37ff380fd9b28a5c5a655ed8b3ec07c677fc1334d9de17ef9900e710e5
51cacb225158e5828a35ab344c5ce17ae373b306296e28a15bd0e71767b09839
ccfaf87182ce6b7c78484e819641f51f77abc578f2e9c90e9adcb6417b99bd73
92dc3e763eb5905fc96938ca963cf37ba8bea7cb5d216b3558231863405f4061
5d16786886d1552970bb94732f8ff8359c15a64e4a4786a026ce1afc8a9cc524

1d5f581edfc3020d1112937a48ef1d7c581c4d4f239da31ec9fc1560b8d6f9cd
233db5d8b57c0f32438b5f23d6b3300c905e9cd46654d385ed10649487d5b330
a399b33ae5fbdeee9485ef03740d93758bf51aaf502d34d9c89d85a7f3b86dfb
67f353b68928f22dddb5e4aa938f2374da18f60cbe567bbba7602ce4519b6dd8
5cff2a0123868066c09002b176825ddfe61f6cb05da5723466f894b5d76f60f3
0b5f5ba1e2a5335a5bd7f8956e473ad9a12acad301cab2f52ac470bdbe889b90
02382f52e6ec20c837810aead041585c1f3f1b8e6a4fd741c376da56c156385c
f3a1fc0f7d977c4de8ca0fe071eef56c9c93e1e127b9accabf2c71750c30112f
0bc259edadaead9ec5f2ff8cce3279cd33d4cfab1c44d4f5a43c81bac57f19b2
a96aed32935afbcdb83aebe9c12530c45aac3a38e25ba9d83ba31f6f0aae0b88
0caf8877a19501a67c6fd3fbf530a452cf5847c111f71e40124222562dfc46ce
5ff9a33656bd46c33d2002b7e0e7caabec4694db5890c63ca13effeb53f40979
f9662afe0b6565e12fd9699091f9704f8ffe2e02cf3a1ea09c2ea001f684b012
7204474b156da5a5d03e7388fc665c5adb6ddc2161a0771d62f54bbf98ebd2fa
6014f7f78826ce14439b4fc5f8f2163a7d85e2c36bc1f4fefb85b214b988fc16
332a9448ae2b17e8adc65d9f89c22ea15b82ce5e980e0d941330a34cb3583421
cb1543966afd57b1d8f7076fa9b96293b4cb0ccec5d55c1b5a5aab2a0f227766
bbfec471aa51ffd0a4b758029e2f627127cc620fc782597fcee15022afb8fb8a
6de1db2d58b43e486cf77baec2f37f574cb76eeb811731bad68670fdb8524082
225ecee04e8b3991931ea6bd4310e16818278779e86a038e8485191d00e15ad5
60e0a7000b4375315748e211e2138cad2f1fb8faf21087b4bf03a52e13d6c9fc
8ae44dccc0f084e05130c9f9fd069d453b24426f4265d155b0c7a70daab5d083
409c8cec79ff314e41367e3213dbe2cb2411c31877b59581c7707b3426dc018d
0ccd46f9cd00eb6d352242c93bd9b9f512060f0ac7a71ede8d81d7c601ccd4eb
96d636e276c345346570be12061b1f8b4dc8c5c1552f4d8123ac976e9551911b

99e85b31e6fba137cec181857f74183cbe69e86a2674c50023d46d53a7081891
ddc5ac2a80d10845bd2669d5c914fdc6d552149fdad1fbc6abc9c47e8d492ddf
8e6fc22728e4d6796944071e2c844401167a9240feb8dd5153a68e111f5cc787
71465ea2b84645eabca2cf0d4a605f55ebe619ab7cb3d533c7f0f9fd6055ef8e
ae87f08fa2392ed0f2ca81f2378561d4636063f5ac6ae0c12ff85bf7cf958168
57e0b56de0bd828a43dfd8b373e281e12caa803f20a8bb7496f7ec9877666543
d19069859676a7707f7c3ec23bd2f4145904be67b741a8800667bbc0007a1e0e
3bd0577bffde47cc5534c158e9e4ebc0fe352ff3cb49045ffb5071a740a68979
5a8939bbdce399d97114c876e88f38fed5de83c1d85c201a66e0fe17c52603d
e1955d41d6c8bb6a1e084099b399374f6d368331da0b7c92acfb43a53139f007
1c92b31f3a469052d1b1995d5d0acfd8a00e78c9925a9ef6deecaeb5d214979d
4a91c6e7e9d76df51b4fdaff2759017dc57972929306404d1abf81c2f2ce4220
2c6a71b4b1ca1d23c97dc41ad8eeabe307daa628461f4e4fb12d36f9ef48d112
cf2a78d02bfc62d41af99b36eef1409e4044ce2a3a3c081a707aa8b9f49bd666
c49722c38967f9c543f296aa008b3bd99709f1ab599e639a0ec295ffc198910
346480e1b6bac109110bb41e169506db7ebcc39114a24dda7201fce3897c99d3
b918459eefdb05e0764c172d889375ffb42c137113824d82a2f873fd7b1014c3
bce56974d66d1fdd0ca847eb25072e5be9dc0a46fa1174340c94c880981838a9
6b3578ddc7689e9f7451a9e50801f7ac48f6cb23d1b19cbecc37e1d6cfa54e0c
b07666833b298d7abbab9edcef93952254fc70a7a54cbb97ef11e64de2461f73
a2ac8be6351c4bb4e23c4f58709ada82defcd8473f7d0c2dbf819ebb7ee5b082
9b9481dc677ca579f0efff5101f12be40ec97dc8d67b0eb18eaf0fbed89d679d
545b63c7c623de6b22c054b6987023cf457ef0fdb3300f7b804d7433769f3034
1171c74aa57390fa714b81e1d52361b8e471f71ea8d5007822b5b589142510ed
e7c5bf66ec7717f2f14eda181d11811095bf42c024c742d6194a2d7fa1f05259

a0b5136523514a01ab5dd22c6ac605c6dc618190912646efc363588edd4ddcfa
7a91dec008e7f9e85582f316a49d160be39848ab894a495b425261c9bed64599
4b0f1f45b50f9d90b8a3f4af1d533984292b16351faab4d505a5375c893207c5
c805e219049a51d138dace3074dbf5bee4f4f97f4ef6b63f91388e98a2d6eb34
3d738f39cdd2141bf5f41fc036862a67e98cbf3ce41b1c3caee39e14e5146a0
e59acf3d3c69d739f13f21daebda34a0e33f04c9f055489ebdf09c84ab48c4d6
af59ae906778cf92feaae87e838082214224fd50252949c4d3100d496e952f51
3d54da48efdacb5393a0f41a2ac94ba8ab2dcfc2c027a792a2448a765c104671
5eb3afaadc5bdfcec4228bfc2e13330924961dce9480f5e1a570e8a73a65b291
85d15616836780ccb7286b54faeffcbbab88253bfecfd477934d9b94a6e60748
409e73afbde98fe78522e2aad938743ff84915b76936714f57022b228d594bcc
539777e89ab2d920816efeaac70391d411d424409179a3713ac8cf84c2170a1b
6a90cd84ed062b7c5a79b729ba6ce905c23fabe927fe5bbce8f9c6fdbcab17b6
07e889ad34a429f3295011d92258f5d43a6e015eeb072695fc81535f82b460c1
f980857202993c2ffa183af7399ff168d7629e9ffc783ec47019d37d4b808809
50fa028368e760bc85d0216e2ad6f80446fe8698804d8d3cfadbf83481ba68e2
c7bed37995e6e0c2d2632d71214b84af6f3458b08d2be9a1b9a6d845b29461fc
920f9cc0337d2b15c1bd07b090267be8b23690d9ec74a837d299f9879b093ed2
909e6a3a60bc50f1633f1252c42b41eb640828cd7c9bccc1eef7750bbff427a6
310db201423dfc6274b83c013ea1970f9cfc98d69299f3f0894a8ab523bb4abb
578b2ba2dff10c9c0489a5b6035f601453f86d0884e51852c938fb42635f2f81
276ebd923344e188a61957638fd70fc0464b862429e62dd7bba561ebb7c324cc
f61f5b33096018b887a2481fafd7e1e1ffcb9439d14886a86fcc8b7c0cbb84bf
605d2cb225dc87033af384e9c71122d932766da2c06ab55fb000993ad0726201
84dba96c71e6d9b5a098e15830bd8226b05d513fdb8fce9dd78bf49bf11e9b6f

a1a5b18a607b7c0a42934a416ee1a9da73bae048597bb93926e41fa9e32ae9ba
89adc6d8ba1275e2de3434cd3c98382acb2a0dfe1b0f1eb5c802ac8a0bb6ba54
110893945252604bf4e88f4736e17c770819cda5e5a09a1a1a1bafa4d67d232d
72f45c482b050d91a8653afd26c4acdcdc093ba92cbe1715ffa017d60fe46b1b
fa1f6950bcd1746e1ce9c178e80fd4883f4614044c9a6589a3a732c3cd1f9d51
a7940c5fa64d96190512b08801ea5a9b2e146cc4c778c2c60fd5070bc7a6467f
949c42cfd3a1960fb45df3cedf70d42704453daccdc5175ca5db891e2a3df70
255e9e0a17feda0822bc14e70375984e2a575e28bf694320f2a4698b8c97ad72
b57dad2837efab8c9774abedc67fe237993fafb9e1ab266dff0d074e1e8f5df3
b817a846c29751d233ca7a1ef7882ce22f13e7a60e9bf364c7cf74a2a6b390db
ae2015094cb651c0c5cb1c733ce9ce51d162b807844bd49d0f28ae9ec5ca6344
7ad192a13e3e001bdc2f88fcd9bcda481f25d99b0e5fb1392adab16c25f9e40e
bb2c08c18d1ddcaeec3f6c85a9245d1be8b8b291c265a33cba5bfadf0f69582c
9da8ad3765e7115f157bd86b762215936543f19275afcc755692f622e2610e28
e64b24765de45ca19ae6af939e455210641561eb20b1c31ab47b98f05a89e8fb
b569291b7365db6297e553e3ca084efd16133c8d6f6e92331310bd63ab895794
caf8e871b88a2f2506f5c3f301b31de2ce5572c2fe247bd109015e08777cb119
fa9c73b1973cc52a4ac70e8dc043a9f1fea07e6c92c672964fe96d19cb153a7d
de01e17676ce51e715c6fc116440c405ca4950392946a3aa3e19e28346239abb
80987c1e1ab9a3e969ff4df39acaabd35295c73cb53983d0492c5f2fcac06ecc
3004203a1d06267765c21859aa9d44fd4a6ebbb6d1dc4742e55e075fa067ca23
a03f78b80420e10e15a82d7ab616db14786d83ea403029bc245d0c7cc3c554a2
a9be395829b64176c4830a72e904e27f27d41bbb9708cc477fb33d79037b44fe
26e099e5d19474fdb7b1e2b7a0136d59b27148add33cf50f6cf870426c4ce772
d9262bbe5edd66e51bfc1f1c9a9a38e7d6ced8ac8204d7afd3d50e97ea048f3f

9037d211a9733098179265f48349346d443695d30a61fa1ad490656d587b1212
0cbf0d4823ac522a7bf6c27e149423ff729de339fc0189331f2480f50b87f3cd
d803c2b2eae242a5ac10d3ac4ebc6dae25686d72dd594ba6f74a8640640bbf6d
60ca09a773d121cdba60d418359c81d65a36b34af8f03f74ba484f8ed93e6821
6b0717527047a417c69ca0320627425efdb90120367136b275a3c4fa4314fb43
8d5a650ea0e2007046e94ddb2714071ca09030ed5124c0de2d486dec6b66155
37208f1cf4311b1c2ab82b596ac2dfe36ec9cc46a344008d38ef688e6be5971a
a09611af25bf6a2c64f91ed275df8b05aa487dd67fb91dfc2dd25be273a037fe
d89a8fa17d017b01443451e0d1daeca4a354aece668fea15d2e179a75e842933
bede37a476ca9c81eb008431625f8e6948ae0ba279ee37f8dfd30bb5417bc163
f28567ccedf08c956f47650212f06724b58ce6fec2ed558f6fd54cb08c1114e4
2bfc7b3413d97c3168651b5c61e6738114249f9570f7702af467734eb0d138f9
d24793207e0ac93b900b15568174e644bd4e696b3b4bebc108802497e4bc654
c7c12b8001aa3f087051119270f85d023c1a60204365b3cb9070b95291758ed4
5f587a2baa19ebd3e30a74d1300e8ccc125c83f4d54b7ce43d03418443ce7ffb
d1a663401c9e0c9ec3ef1f9fd86a2fa584bbcb2520327aa72413b48bfeea3441
8d72c2cb85ac1d506403594695a95c6920722b637d02e351f742ddd73dfa2e41
c2db4ec235d7bcc4c7bea322ab3467812a6f0c0d3e59e81d4ac5a4f42d325413
69edfb84497578c27092e71c7591c47cafab35591c490c88fbded9af5a8f9c3e
4ed2a2c86b4d78a6a5a735889400157c60c76ed087c156f76faeebb06331e4f8
af191d55c272a87f32af63ce3f303e37e9e4f5a8fa3c9b0aa768e04b32c86e45
86cf0dc105b3936d23f4f35d879493dbb39a9c52c017b2dbe8bd7ceda44bdd82
18e333aedd9cc2443504b58336c7a48ffcc58ae96b119fe9df5debc377489a69
895ceefdf08fa67185f5ceb8ae9beb7ea34a9e18eb5d1d83138387a5b967dd88
0ef4363b91fe0bc3768efbec01936bd76fa22f6fe9c9f09b632ca9ace39c4664

b1fba5af005cb7c1c2769310a534eb00fe8eb6536fae4a02ab309012320bf94d
413211a147d158f4bc02db022e6206ca97cfcf97d5834030be956a6cd7807462
8bb30172570a834786fe19027cfe27fbc14421f9e335756c5aaf43b15e735c80
be6fd5540ca4e0da050f50ec584b77441e24075743d338a80fbe7b14ce11d3b9
4f907d1d00ee129d28ea511c5b12fa1d926db035a93bb56da54c97da446a7ad1
8c085560c0f7d6585ccb1b5bb21e809038c1e94081f7a282855da3b80b6d3bc6
aa3326e7f079ea9910a5ed11641741e4de945852b3704c04d4968f3f591342a9
fe80006844faa8f8abf80fccf21465e0e2e43ec3658f043610aea27516dd5fc8
124ba397a87aafdbbc11ad59c3c9469b358dcb3c30852e7c55d8ce0f4da00d5f
8d751ebba75b14f55ff50e3a04a4cd7f68d42f4099404d3ef135434baf07b889
3229ec4a15298c88494e61e42f697e4057ee1a03d5a956f2f158f6ae7189002f
0886beb4a2cf6c9ca558ce3022a2345772fec71b59f4b7952e97d96f5774e28e
158007ca0402e5798f282cc165d16972b2b17c8fcc1bfcd8c83730ff95b0514b
ea80c34ab0c4023411bf8867c673b021a0f2a1100c3f9f1ba2f7feb7d96a2fa9
fdd6a48c9f5d40b295099916072b3f4323b88e5d6743bac55b7ccc3c288ab138
12cd280cd3ec946e791591ef52f5540d2f9cd5fc1e53b9b796fca346521862e6
1572cb7809605fffd2b5f231c0bf113665bdeb6d26d21566a833f275c99f10b8
3d63ba953d2bc769d3c4eec503898b568421a7399182b8bb88e2e3318f22d860
d315f064a4ec5fad0e4fb060749060bbc83f08555cfa791adff31be8104781c0
dcedf503d33df63b76ce0f1e073cd662a9e603578e2135e3ad485d5b1b3d3b80
6af44534876f15e4e91489d114d3da3016b5254cf55e13b5cb879d69aa779363
8a169dd4a1081b0d31d721685393681a72e7e2f371e4489106e2e2620abd5667
b4e6d12366a4cb74003c75c6b62f6077b6e989b45ca8dd340bea4bdb20782c8c
fd213abc3409af32d0bd34b9771483341486557d9770a21e0c9e95893f336e1e
ac261770ed61e6f38077475016b53535cc508268d9f4cde7ffd0aa0f6756ace4

b9e6a4bf58116c292333fa7352b841363afc76a5ac42f56a13b4404e6a05a2c9
2c29d4cd67cb879daca1783e882edbc89a1861dad1f5d37db64c2c3ae90ccf2b
350960960a32ccd9b26220b71957a7591a1fa680179272ba34316acbc62657cb
59677e35633e7f0ee48f33f562bce4ba82747db77ee353b445371c6ca0ceda82
f138185750e5578dbbf7fdab1fc25a17e75f0a5495c7c2eee8f20eb0a0aee83e
7bfc2013a4348707c9a9341a188ab4470e263ea13d9ead4d581a5b01c7da95bf
94e9bede2ad4d59ff789b75d1f93aa993b7bf211c1a2df91c1e634fa121c18bb
7d7ffc4deb8f6d1ed13c7e057702bd2d1bd5840e548d015b31477b1afbec6c75
6c08a7d2a718f0ec284e0faac64b71609a0f278f2de75da84179e98b6dd81717
a9f8df1fa60307e31e0ab3af3294293877ac53d9b5f18519ea2bd53aa3540e1a
19a71531ea0323f89ebe422b0639d46980bc9d7a80516bfb651a4ae948d8bced
d073eddbfd105ef491f8ed62d340032ec11ab9928e27a3f166ae4cbdd829b36f
fe95671cfd2d940e40a0bf1ef6a5464f79553f5bf48ce15296509c7d6de185c6
3f62fcd731288b6905dff10c6325d794c4d1f6efad8085a5b07d19e6d3fc7d1d
2a20f13f1f6f1f9c34b1d50db884dbba58c27429727d86705f6ac617a2f35ba2
66614cddb018f17d959b8aa84dc606d5a5a0fc61b0f3666f152db360cd25d3ac
0cc3a0f8b48ef8d8562b9cdf9c7cfe7f63faf43a5ac6dc6973dc8bf13b6c88cf
ebbb7c2362a0e6419c15a9308e0926aa05273dee3eaca8048fbb695770bde00c
22889600bd08b0b5bb99606a746173d672a1bfbc445774059e14606720c135be
68af7a9fd9018c6f53056b3efd207735b6aec5f4aec35d57c81557f810c96412
65ef82a20155f3211131c9b63c787e4ba5d4b22292a92b8684bac2826ab3fbaf
cc6d59e8667f343e25da262e6b25c3797c891c6b56ef797f9f05b44bc9a58e40
5b4f932fb4ff3ea716806ade6b9a664cb856e3a9b055b613a60504d4e01e8e19
58fa775e3ffae9fda681b2c2c1c2c48bec7992a3ab3de67b62f7a59fcdeed62c
127166d6299847c6bc03141eab0c91cc33a7825b1b3385a54efa24f546bc3527

035c46550d01673b6c56f71a4b92f5e846f1b1a2d784e0b1e3bd569136368792
c2e9c1b71769fe72c45a6985ed44994aab1b5032d085f75a3ec93abbc4d3922c
c930b7b389953de4a75e889278f368fccea44f435c499a154aa692eb7847978c
ef9590c8b0eb4e6a87caa491ffe5de99504c295a5df217153c3fc3e9d1db046d
cd66ed333cfc899e97c6f3c805ef2a1875e91abf29f3144d629e5056bcb0624f
8afeaf976bd26c17b8045ced0b8a0b39198e9f054a3857a999b8b5b11035698b
cf6fa8cf292f5e5c907502fe824f53e9a5cb935d522b26c87b9ef21a3aa0c02f
a665e545cc4500e701786eb2f9379bbc711d6228deb7bd73701ff3c8a0f16a9c
ba5a2accbf4fb714291ced8591c8927c3d124715b11a74929a223486a69012a0
cd30e7bdaac744d54166f266000569a86da82067e725f52cd4b32252347231b5
d8883d2bebe36e3734352579cf9259ffc18e57e18a6cbeca6e3f16897f577dac
84dfaeaf05f38e93b8e3caf9f7bdf7f009c5e6b52643af42e665173f80d722e9
f4965a66382b8e08dd570856537ba311278e6a677b62cc208b5ce6c868f33ab6
7f2056739d2bc4025ceaffb6c11bef4f826e004c8e4d1a88b0e862659f2573bd
1e68b12335cbe6510fec7329c168486aa59c4afdbf86c0986971593b9c445c42
43ef37f58fa0b87cd479cbaf8cae24a2686ab8c65ccdcd2c544364918226142
e219c7692363d60c72f0ff86fce93ce17a4de8b4487299aa247ff672a23b51bb
574e344842491593493e45957d0824fb3b1b098d43851c5a0e2a68199c4a28ea
73322e078cff40fc307d85c44373f12fd3eb2bb79921b56398250a2540ab2041
3f677cbdad42987fd876d343c43facefa8899927bc87e865308cb9e5b12c31a5
78dd77aeb8eddd08ebcffe6fe5aee8adeca93cf302c60f9b5be94e63532fd1dd
bf19d5753ca574c1c28ba54ab9697f1b32e9aadcf02433a923886802f8e03b58
78d081cd2bcbef77d8aa2f346eaa372e008ba08805ff5a97a78aa720a9b6711a
3823aab67bb0c2a0654844f337a63404ef2aa8cb25b113e9279060f54582a2cf
eb801ec1842284fe27fa857764657c9f5b8915f17ae1b215e7ddd935df99b37f

91e2ce4efc7bc21c5949aa3588f794843618c034ef01dac57e6c0dbda616b9be
9a5d781e1a55fbacdbbd59abeb9c4ab9953e4080d38b03f7d834a932fd195f68
1189442df4aab025619e67016b3fef7c8ac5561ceb4b1d1a6aa41dc45e081350
a46ab997eb9fe71fb92103eae13b4a839409c8e92fe62c05cff6defeb9d7e63d
8913cec6cbe3ab2117579092571ebedfbdd0920fdf8f738517ce1b2304b2eda0
102ead4a67e581826e24344ee11eda36684a416e0859ecab61d7271d410fc3e5
b542501efc37d5e924f1b3bdfa01fb22b2b0c30b3e456bc4251936eda9c04751
6312e6c96112663887d6c645ec3ac28eed20793bf652b1b047b25f120f038839
374f9cda82394d67079e07b9820340a8b0b56d8d03aee0d14483112c79abf054
cf2ee7eb8b7752ee67c4ae3841da04e42dcb3f5fe7070ccfdeab964828ee6f4e
673813aeb8e8eef053a66a17feedd5f0c1a2a0b4a40b0cd5bc7b131f6d26c92d
89ef7867492a4fc60f721c0049f390dac6571b56587304bc9e114c7da5c76776
359b8a5479719b13f0d91d7ce6b7000bcbd399cd38a966313298b65b0f200e48
6953311741e525d87e1adc4c059beebd651a9f68b372921b2461f18fb6c58d36
d6aaa66ca79bc49452a184140f4f08c324f84f38a3087dac3e44b67d426c15a9
4ed8fed31a4d279baa475a137b4706270ad33e185514cd50bfe91da8bc2c85c0
587ad4c3addbeadd320f2264dd77cfb1ff7c26b303920c4f1952961c61dead65
1d387437a809ad69c618a90330eefc6b1cf42f2c4f9ff325f625387faf65f1d4
273a0d734d13943e142efb8ec2e2fccbc388d184e04c21f9fe09c2303e8af841
96204087db53a1d8ecc186b10346f55b89560b4a4ebef9db98b845bfb70ed3c
063fdf0a15ceaaf0acb843c34067b167e88d86bc27e88671a007c5fd22308358
65536b9cb6a887f0a1a56183a2457c0325f9757a5b3fa0406365e6b756bde482
fc81d9ecd330dfa4a0c3b673070a836dd95702352b3d7b9036972a8e11438b0e
22c1861ee7b03f18b0450b26e950cfe71599ec7842f768577da8f92e79d68581
d561ee4d6657d9e465885281abb1589efd2828d47f5033ec10b04a85446f9930

15723a4b0e16cc0b8010e35ea3d4bf48c2c291cc247821f701ed6112b2aa00ef
22eed594d424dcb890e53916fef86e80d36833100c8dc605af894c52e156fc56
bc3a7ba547b8a0f5cc6be6748eb9fa06ae2d09ca4b3c158add5e4868197c72ff
8d07a2a48e4c8c3adc8ce2e5b6c5427c6b9a4c847431e208fb52aab51cde499c
2478d00c1845fb381f4a95ec1b5e7a6fc06fa561c33b4d5425f4ae6651038f46
aef2f52ba5434a78a7f3a8d0ac2dce45f17af8d9c4b8574d83a46b254e3e501d
ae89ce969fa27fb2387b64f721c46b4349ae6671e4ebe0ced9d6a6b369dac25f
70b9af50319b0ea9009f6cbe5d83bbeda677ae91427061b4a75ce0586c217cd9
ad00794fad5f05d585a1b2428e6ecedc0ac13dce01019f61e186fdf870086321
9827bd562f66f35b15526b9e51956d9602ed5146a54afd70a666eee4b8b9bbc8
5fb4aaf0d1c542fa8d71f3ec630f5fd64532ad985d1db0d573ef8ac7398a98d5
0bc0bc3e5b57c08340f3cedd0ed25c8c25762bab8ccd65c7b5b7f791c504e798
87df98ce409ea7e26bb37979820ee2bfa53a3e1b9a06e4aae849e5d7235d5026
edaa8b9f5332aa694546923586bedf59072a0c4393197816cdd2e560cf442030
198ff25ae64e79d730be54473556ee0e6e5abb897708512838c3e609e0685fac
bc78ec3c51c7196cd446f68e84606d5c3825162e1cad027b0f23603ccb2fc51c
66ac5e37b797f06aaa4b9988fc253728f78e2616d39c947d5503270940008661
1148db3aaf5a05a4417caabaef319be91bae987c86dfec8fe4ae915d754feee0
6917f594412e3365a45c89d03c7d24ee7655dcc72fa2ca5d7d363356fc463d15
f39f8938e5b5f0f8c56195aede06926108df6f3b9f68a43e9018b6e49c5db9eb
da922487bb4c37a65402da371e3c24a42ba05e5b0168f1bfe8bbf483fdd5484e
3dbc43f22c9dddc690fbced32eb230efa42cec00bbd6a76ad00f11c84e73170
638155d89afb819284dc3113295f1c89b09f515d7a2fbb4aef4a826d952aca5a
4f62bf0f5879470263297acf7b297d9b68c12c6422f7913e4822b08de609dadd
a99138a6fc4f44c76bb021d2db24bab1bf4f3668c125d8b62e4dabd32a1c1252

10eaefe9ee391efd3b36fd4b611dc55e8a01aa0eeade1c6fcd1dccfef2a8f16e
faf62974fb6164d592f0df8df5142e648ba557c260bb9f7dff95c1f6fdb3e62d
6c84b962d87c079e47b2d713abbd83d4d84141de5c63a56ed76e3d87ddb2ac08
4b79018c5282e446ba71fce1a1380b91c28c8171af70e6927b881f5bc78b2e4
9997d0fa518a0629fd279a263f4eb2a48157514f3e5b6b099b5b781296190ba8
1095c72395f44c4ffb52166b735a43a73b850c207c18911b186b2597cfdff1d9
1a5fb8d42dcc121007c4c61d6c04135f5c9037804b91693fbe6fa209c38dbab2
c06bdd6716fd331021d8998b44d9564f6762a548cd75454ccd47b3fcd77c25b1
a79cf98cf569ef3dca2cda29fe4d3d795323ff04830be25597a7ff0459b616d5
1d7853c8f73f71d2cebb2b7051db9cbe960fb1d8a7a11b25ed07bcac0691c2f9
622d3fae11fb72c1723959e13d72fca40310ebba71814ff2006f022495e43642
d5970f8a7a1af31a9b232838e478f3d6a992399e14d7ab49cacbbc61bd263f68
1eca8af68ae800d466cc84c35617833c19927ba7798f50e3ec1eb14702b70d94
c7476e3171be452564ce80f4ad32960cbd9d0dce355fe64a57e5e2a5b8eee553
4f24169422c09baf830ed1d6798292ec8642327d288ddb9c1958a460db54d14b
44111d6aefbbaa8fbd03ca977256190fc2fc6494cdc88143eedf5fd5a5d35d5a
a4e2e0a936c0b39b314798ef0f29ebd28abb5a9f96573ebed9085124dbac23da
12e51da472ef56ceddaf7003c426471809f708b3ef4e8723c7d917a0b2b5d751
c68933a150a3930ab10c71e7b7d8f6fceb700e71e40b0db02cae1645a31bb81d
35e4cbb575b206562f8137f90909607313c62d93ad132eb15e0aa5d132d4c232
a7917fcda306ff6e8ff7184b426d375e07d87b80d67deb9b78098a09870c7a1d
665e86bca353d50ddab9131f4363f0db9296a2bf63aebcccc527d93a78b0ac2d
81fc7e8eb634f4915e257ce22410e812aa876111a275534dbf7e0f390ea3785b
db8b3cce27a2c25f0e2a7c60b52fafe0543a6bd7d0efeb2c4b9dedb2950e3b9a
e5cc3a637bfc6c43eb98c9470156cb59a33d3bc4a00f7f41c6fd7a9090d4259c

273bebdb90b68699e0503dd2bd4e798dc796ae237a3b712d28b6a15bf159113c
52d5b1b05b1467dc5e4ccd5484d7e90b18c10b5b1c3c0cc80f91a4f6f468ebe5
a4e251c7a6916979e4f1a7ebbb05b229ddaa172854e7e2814d67fe212ff4d74e
18930277715eaa1fc26ba214e71ba5c45b9da8451bb56487843e3c6b186be2cf
1e88cf5f731cb5709338c474340dd1e49158cca9dfe86239c83bc64f1cee1377
7c65f253762292b557bec797c24b326581a933216ab2fcd640792fa588b14190
07dd866081de21f9de0381b07472e1e2a9111fa537e5a3e4b4ec8b41bdeff509
b292148c8a2af8d19986a005f369b2486520e8dccb563d9f773431c86730d76
50cfd43c89ef41b0110db43fc58bcc3896ca5eee6eaf1147598b8943a8d10dd8
7902d3f8af71dfaafec2f022a930b11917d999be97fc5cbea666d2304a9c573c
4be97e27e3b869ea728c0975a0cc353d0d4d09c4d535af98b6a4e15fb0b6363c
c709f1fb9360839c290aa29410fdcc18533b2ca3c7af78b7e400315bbb339022
d62d827e52c7ecfb79a6b4676b8ee86a97372941dba54ecbadd54156de352dc5
4c33beec45cdcb37f23a240d2799662be2bdf3894c126c4fb9ac98a2acb253ec
071046a3e8f34c355f0c0d0f636b53433fdb324a6e460a6dd459df7b101a9c2a
7baafadb775ebb4d97e4021a733200f784abc1fc6567a2a755f8e5d65c77e29a
abb67fe6cdc14872138e8a32409f85755a7e4ac9b52d67137efc3c2fb8e932b6
08429d07efc7e389fee4716bde1103463cea592918d4ef49cc61a54e4657d36f
d3beda27b66f9f5901e2b3251f1007cd55eb4570af3e9f0446acac8fe07c7159
c77a64f262d38eed930d8a4e414b9dd9249ceb276275a9f8c21a07f3beb4be9b
e8cfdd9e321becc697464febcd9727beae99cf967517467859bae5ee2d761ee
432d74b0828e930173c8d853518c1ea39a91a7d9e805fb436c29fbb9a06afd2c
b7468fdda8a4e5985f625fe8328c227b059e1263eece4bd61e1e3ce7122083d4
a708a261d5e203c7c0c789cf33f251c03eb3879f24e5ee3ac8f4f7c0991a895f
e4c7302e9fac79e8b33cf89ecb038139011d159148592024c2b696598efa86b4

d5eb6447ce72f44bd58e8e72aee1369ec2c5634ba04c0ae97493b4073127a395
13a6113875ce9745029aaf46433a25ddcb5e7aba9912be66b09e84fedb688f22
f4a74294fb587bdf5cf32d2d58aec90458e13c3cdfa2bbdd2ac14722e1c5e4c2
72ed9a6d3da7acd289de898ea0657b9265767a8659c29de1a864b95aa501c232
f9386045906acfb2faabfe06fe4e0475ab7f2dcdfebd58a2e40b531e064fb23
aed04c910be5854a9b3fb3feae66a28d18898bc95b36a86640636a319c53f280
2d1f46eb0a885e40c434425e5ec5d741d17f66abc3e63b646b7da53c4734d881
ddf80c905896909531e54e2ad3a748e4b01f7baee199b79a182e36fd78d16d74
9b9271f77d3dd2d41dbb300d64f7896f1c16cac30ef3b0d89c2f71f9d42e648d
0464b64028a7af3bdb024a9e404892d0faeb3dbecad17a4b998d062072364f36
a6d6741666bdc0f5901c7023e50b2a2d45f692231cf179058126c83aac7db207
f053bb023056264d68ce0fac5407ce980b90625aac9a164e770a26d7c533e108
2296d34687d0684d6e726b6f5c1aee87993cc303a9f10d39866703461e48b347
8c6c1fa83245db5d0f9dcf51e55cef022e2c8083447f16dd9eedef72a29eea98
3c7676a6ac978a303b47a478084feb497adb29ec2693cec3f2e4135e1fe4a974
45972319213555f2f32b95a6350e056ae4c3067026e0f872cb145c607f632987
1d3c0c4bf43c6f12ac606378af3151e9ccf5ba325fb9a40aada9491716153007
550935dbdd4fb1472bcb126400b9305d0904c79051ab450187ff48e044c47269
d92b279e2c085e66a87f7782e14970de5e2c74afabb0efe4fcb6082c6c488830
1f3f1f885e8d2572e3c3804b703751d71a7017b0528d90ccc3a890f905545fc9
700b402a1229a1a7ccb845db7effd2c3dba828027651fe4a83c2f32349f3ae5d
97c6e918c3df984783e24ffc2819839dc1a90df1582207741785b569bfb69294
210fac78b1b922203250e7daf7954c9cf3925e1d671843fb10f70d1270ce45c1
8274c619dc8e529f1f25f4edea6fb3851f838a700a6a907e0c757c69b4487128
6282ee51fed1424b960ee49c62c49d9d9cc1e3542ee17e8ee4aad3234e786ef3

485b659fd86857cd822e59dd105c80a2c8d7584b09c56445da2cbbd2d2154c58
18ef4eb061c3fd5083e3f35b0b8b5ac595219494922fcd12d3752e4a24c929d6
41951a8b6b90ddf7be69bb9c033facebf7d8cbf3fde2336d9a5f1a867dd9eee0
226468acd12213b63dd8fd0f566698f824884a7c919d544632197392f7a686ba
462628fd86b0db2331b12e79c6171000bb4ff4b121fb59b51f514d99c1c82550
44bd1ca0cdd829faa31ab278e7524caec07ce96f7edd12d720390118b3ea5570
c378c9853591e091014eae114db7830c960759be5aab305f556b722d71cda41f
cdc30ddbfc12727cf121b59e058d06b916e59010229ea7e9fc3f9e604e291018
584f79e14f43c65ae789887829e565be4a2353c98c17b1589477de973d8e0b1b
0fd20127d901c884702367bb04f43280d66fb10e13c630dc33d63859a68bbb9b
fe98286cf3ecdb07fc0d14addd5290484e279932bb380459dc9af0f75dd6adab
0701b4203c62c1cb80c3bc16135ba29aad68d16b18f153ba4c5f3fd4e3721e30
536f9f3cc93bf32be8501fbccdaf2747154e5f225e28798795d3a58b01aad727
4be4f0754c0f17f6526dec3cfb1575e23c22c82cf476534c7a6e74629541fc7f
615051cd3070bea7abf78364e9267e1717aa477f7e3970b37d0195bf46c8377c
4c6506449a6b0745a53142403fb89a6847a38605429acba4c2ff1d650581ef1a
7943e203a5b491aa668e52fbaaf28a0fbbbd66050be793aa693de76ec75f2098
69ec521a4ec400367b6c7eda23f59d642ee4dd5f2f315599891eff067ffb9646
20f081c2959fff2f00ee79d044b69056aa32e4a8b6e6413edbdcad298e991fd0
4102ebd6c734219f1cbc024a51a0e298f5831433872360b23d3ea31e23a1a119
764c0bc7476a42d2b4519f21a1ab98a60cb7744ee15e3c78e556a4921747c4e7
6c71e2b4685fead08b71f970e8320af133fa403887d2579b2aa2c29c9347c6b1
ce321aa2f3deb4616acfe968e8a092816bc43a7018ac67726be5eefbc717e1fa
013dfc0791c5a42f812b81afbfb5a133610d44acab492b127fe16c94414cea10
5671ec9ba4617880bef3212980e798c07ace68b40f19f5ecb1c350e06aabc9ea

b2e8aff6ea23837daa71c9ec70b1f046f57033bbc8c3dd9c0efc8950bc711d36
a70a9d8ccf4941cd991a1b3d53d0bc45e21eda19270a031ef3cb58f350543e27
019fb224a1d7c9a989a1fe4deca85a36bdd75103467e9146937808015ea51ea1
46951e1fb34e9a26496f3ebda46115c984f17bf20aa41a7149e1142cf5940c4a
cb3822da876ebe6b41b1d9d323bf997e72ade76b762fa2ea545f49f690621b89
92d74148a3eb3c9c731c0653646e9aa6a7749a31b07e7dd928b039adada269b8
c32f362b565d9d823514dc4b512f78a345ae3debed65f6f0a2a8665ba929724f
c6a1f009a525b700837fbe6e0b46739a3c091afce39a28c6f4874b1f9156efa7
437a2f59dbd6e2a5bd310bff74c5bcae3e0ba07f2f55daf6d7615a4f9a37fb51
82e4f2ecc124c3fb3e9ff6d316d9b5de7bc20074c525b0e65bbffb6e6fdc8c45
04beceba763f24f0f4e6bdff12b14332b9589700e897354769350f8c8d9e3045
b282f15f0f0229c752f13d5bdb4204fc1e122c92569aa4c18ef82956742ea002
c116de17bb97d46e8596687a6f6812f5d2d9ad600c48574960920759347dd8dd
a70b4a1496399ff4e8093310d5e72c939ce25faee624ebf4f99c4b0d0f0b273e
08d80cc23fc3670ea4764156a75961e81320c0c51a291a336d819b732f414cac
361a99afc38ea7aaf04ebb231c4555e4ffe11d77db3711aca6f3f5b747d7ead7
7f4331f37476f0f700b7f64e2c23fd61fb5cdaf1f24170cbf924e40ffe23186b
589e25ff06fd0916c99c6ed17dca2c1d191b313448c1d04c62b8b951bec229ba
fbce72438627da5767059d2f925ac2a318283149c77cd507a7b82ddb614fc6fe
64b4fdff6a88ebf1ba203f97e6a6d0a5428033bc68dbbba82a617b45f3b49dab
49c357cd4f7c7f6e5b2d3e19b92535c914f2e2ef5404d07f9f0bc44ee00cf48a
1c5a7b286a452bfe4ca2ccf6f548260de52705608c93c7fe49092ed8c21d50d7
95af390c2ac66a71ef8a8b8cfc197a4139e6a51180cce245aa8ea975192dc7a6
0e5c367f5a20d2ece4fc8ea0fc95a13c851ceaede4907d83de34f8652bade9d9
887372e109c14a6eeb36079c2e75de51709daa3866e8e7d12aff06d96df07a2

c9d644ee27b8a2f7dd30779a427d532cca2b9d1d391d0757dbf4e4a33edb50f2
1260a52314397b47c6ea095f8c5d67a592c566487c988cd01e2e3cec0c1df051
696b218587b880dd46b4227bf2fdb9cff5f835e6f48954e85fdd80ed26f3feb5
bf80eb460ba1a75ab65eef2d41d865c1559988490fe08ca87ac0304a9a4d0ebf
49fddfe014e432902e3fb76a2fe93e2c4115126b9691212c2168483f7e7f1983
49e3aa5c235d616e850a4af24901e45169be3307941059f42799b71fb913e8e4
c9296a8db6fbcdf8dec47d1748f45d84dbe4e0939095136967341f98d61d1d5d
6772dc15d8188eaae48564e8da85d484d172026fa799f045d415a113c6e504cc
50ffeba0fe8424616f994ef6905c1d6211fc5f67ef5a91e696a570a51503f305
17bf88f21ee3cb175befc8cfdbdee08e98b0bcc27985430eceeae8fd9ee1b1f6
cad4c164e7a6d8043eaa53b5ea8e37a91655100518cfde2db8f6e0ab83456047
c3594f18225b4961db0e30881691bb3b8408eaf82f6fe6e4ad0d1968dfe45c9e
e5a402fc5d2fd3a632f882cf092bd34ad170b0c1c8f6c96ccc97cd5f42289e44
a3c4991faddf84cfbdf3e12e3d0cd1d0c2623b2308779c22b841c995249fcd7d
7464d3b26c62e1725996f7e67a190356b30c6c95e06107b6bd654e0c4788a840
0d8f71a387c3bac359c64a9bde69b5704440ea5feac1d7a32090d144200648e
96ddb513f905e4e99f8dc09d267493acceaf9a07934f633c1d63ba2f8ac153ea
c4b394e3465ca3b06b3a393a72d07ec3751f21322e57f644e8514308696b63b2
5ec0797ecc28d3237425614e017f11b9681479e7b38db4860151b00776802294
08ccd2a4239993293c1da5855542d8256351736146169b1c3acaf1461094f1a5
4901fe6e2b0dd3c31b44cdacb77da77a60597d4e3f8b1f654238ec0fdf4991f6
994c80b9e8cccd793084e142f475b80a287c9878ccf374ffd7628b9ddfd3f86
9eaef04a038fbef59949ebdb5e6f2ecc4d93baf9c83c7bc840506ed5b6b044cd
294e93207682d4ba5ec8808475d38c1c6e3c0734ead768626081f2f70347133d
5bc487944fa142ed7609375271f11dded8330803ee868069071b3a613c622eab

da43c627b1d0c24d11c091e7f4bfb2510eab64e7a9b25795c46c3e6bc944640e
59d402fb6eed40b6c2b78765ab89f70e8b2c859768e552c5fc033a9ac27bce98
e5d307b38dfbb210662b249196163f630a27c024308c058364b6d615dd686af6
5c45111d4ad85d6f00369ed5db67f08b26f72a45f0a4c369aa9a53671d10fa6d
87e167b3ac137078d0ec0b89937e615f4ed8fd731d46dbe2f4a7d6a700aeca27
50e547042d91f02de4c1e48f1d0c372112a645292a7a3481ffa50b8d42306168
1fcc34da13cf257a5401b94d9e6a8e8b57ae7d4019ae8b01190d6ef887555f50
e50df8043eb2ea3022f5b366812ebcc1a7bca38eb74fe4331b5348429ccb1643
c5d1a0b702d24dfd0f01e7a341fd3861c85733593edaf1b83d6c533a016bdbc6
fd2950ed68c371c046eedea5a42ff08eb8d92896791c620643850ed3069ba328
01d3b8685d9bffdd69496ffa63fa00155e75d3d295bec50b487e29060106c983
09bbf92b68d85cdb272f385dede876d7566f21f1229373812b1899c785cb0a38
21bab213e8b680c73984268a1cd7b3826b91aa4d3d71e4dbf54ee224dd9a53ad
51c03995798adfce4355e077b136d3550251562be3150f638f0446f27f024165
5247099a7274a0c966bac76cbf2eadfbb5469668e9826ca441b205be0081d37b
5382f12b64611d6094d9f3f323f4839b02a41666e99c48295c37417f7346cada
eb63190af9e5d65ec8d0cb37ab25a80ce1c3dd0bb2210afbbdb51a401672c4a1
7632c451c64fab7c82133c628e5885fb17df696a6691b075572301bf99cb874d
1d6008d6f20fdf2684734748c278552e6d132b8b4d2fda2418c7aee2214bbb56
35ee3a10a50d3b4fd40b3f4a3951f352b3942bca6e70039318252f94a153c1d2
06223680e309029b20f313fbd7a76bbddd8cd6a3c68854e82a1e45a3cb324beb
99fd855161e5ac7664552d386c7c2116d4c9ee71864ccea26a936f5c1078f55a
8913f0d73f2451b071f5cbd7c0f5219060e58b88ca5f127281a00a4ac6003420
2389804fa78725a4e34bacb0c84fcf545b42c53afbc67f27142756cd11349b16
ff98ef2ca94ca12bb7b4871a185aa0f85aec63de7d267bf3739c50d104e040d9

738b5b0500af55729079145958bcd4c9c043c7e73a43cf6a2c714a0bdeed0cb8
bfbae32e07d71c120b82252859caafe521d715a8541a697db6ef15c6ffcc6a81
b9460ec36ce4503e35dcb64ac866eef259923f467cae4f0a94ab7b8705d5abe0
5b1f5e569fd959ba510fbbc89a4270f6c828401935a73e9abe98592854141599
2e305df3d35e7d388e6f1ec4721b872aee15762f41503cef62d8c9c7b41a6caf
c5317bb05c20b4bce89ca9bcd4a95eb824badafa1f6bf4deebbe9edd9b67fe01
1ec2e233fd9f3efa46bbd868cdcc50e37cdcbf7c9a56229a9a73e7da9cf5f537
66b9fbe8c0cb2fd04f33eaa001272d2454d19c83f47f6018b89e9ef255881b0b
c3e770ce902c3505ba711572a29f7009a865b94e90c31b864e7212f443973f33
58f9ab818b72b4eed18ca05f72411c7138f63cf7002468943abea817dbeca92c
e5923f3a9d307bdef38bc1272764163144ff54b50ba9559616639852a7e5a8e2
75310e5f46bf865166745cab4c59908a8776dcd702f92dc12e6d4c4beae275bd
8e9de59abf0497dfc1399d18c1d41d52c933678681d673cf68c26da4832ad084
edad77792baeab71271da3d6177a5760da5fed7ccc06a05efe16296094006ecf
5b38725e505922f00c3d5a3af2ed0a136b9ca90545492d376958a40b0998d05b
36e5989dcaeccb2780a8baab651fe772996db8d957b261d40fece82555ad5282
4da25e2ff6594efd2e5f71e6da0ed0246f161cc177821257fd98e25237e84127
9fe53dd5538f75e4d80e4c1877212278f4357626b82b52db0fabd551535feba5
8df40e5a302de6e62c3f33c030f49ecee6dc2e1a7c7c4ff21f956b7b35a141b9
320b1c481c265f3f261f3976a2368d48797a681cbcad3437420aeb50a5a7fd26
8da9258c0fcdd2e58f8a558adc6f9f587a6aabd5613827b66201f39ee3ff4c13
3ad835871dbfc16ab05f3feb34778b49fdeb55fe7f60908b6aa4f449566ca609
09191b8aeb028670b293d0446000c70b7c3616684470eb03a79ca4b9184bacec
c5ca71df33e83f8a2012f8a9c8ca112306a3535078a1278882d7e9354cc21ccb
ec68a0f9923e5d22e01c6f0fe562318caded81907d0f07806e75ef4228d2a1ef

3600a8d5f833dfb0999e0905998116af5e64e39b3e48d6ba732635f218799da4
e5e7830b52eca53670a105a59a4cf9edee77f1e14cfbb945265a970a4ff8332b
0120cd580471e2095397ec78c2d2c9a7b921d30367bc9df6da959bd0c8083dfa
1becafef332882ce73323d263fe38added093312cbaa9b612f99758d1dcc7961
754e5e743cd6b7e5f46db32e5bc0ed0ac9598972228f7560b568153a418ffff2
0938fca638d7b96fb97b4346d084b4c46fc6c38fa451db8e08fc1a699eabe7c3
fcfc6c9d2be7f33226650d93ffad0babe986f5f3f342f755857361e7bb051501
1efae5b6cb92febfebb214e2e614581cb402b65f16aa93ec3567e32a9951d83f
5d0de4a2407ef7c1db44c829b8d36a4f5ef625d647c6eafc061978e64d93ad1a
13b3833a24203f82ef3f58570c5143bc40d1f39e7abcf7b34df3aa8a810d607e
119af5cf3c1cca2f1877648d81569a72f850076b8f297a773207dcdc64d6de8e
80cb1edd8bc62c97236da0e1a921c04d9893ba03b9698a1710740b14741b7de8
4783fc4f4ed6a876ff887fc38439c73dd43efc437037d03243c8c4dfb198df25
350be3905b617fd3c9797f4639ca8f1028638299866c16229779a5a97e226db9
8f89153fc26e75380795a57e39456803d45d90ed13a6595b85a3ba3eef452261
e222e65bd07eb962a2941d05dda961000c2ec3cb56bbdbcd757fd67850737406
ea2cf112e4a740b9c462ec93b9157c9cc5f86ee2bb490f32a7c552d4411af6e1
e38026f2750b8710819c91729a589c6a043721a5998766aed363feacf7550a6a
4d81fe33c5ab8acd5b7af51e1f3b853d6f071cc1bd7571b822dc9f4a47e4b67a
6dc0a58a23677d21cd18b3ddd479ed3f212148715ddc5351e9d1ab82c342caf7
97f739acd32886d4a0df621839b6a61be81b54f230f12c93f6797dea3a015200
529d95f398f264a13cb8a0d72e5c6fd6e1c467d7c360c19d75eba956fa4fbe17
7dc906a1292f69c2cacec4c0bd27beb96362a00883ba36ee4d5192b04e0bd97e
7690fa1beb1ee32fb688a6cf69a9c975713c128c65b65424b65256e8b20e4c93