

Forum:[Hack Forums](#)

Thread: [Working with Regular Expressions](#)

realsunjester - April 15, 2019, 11:24 am

Regular expressions can be a very useful tool in your arsenal when creating hacking tools. You should use regular expressions when split just won't cut it. You would normally use `String.Split` when there is only one occurrence of a string in the block your searching. Regular expressions is great for finding specific strings in a block of text, and it can be MUCH faster.

You can use [Regex101](#) to help create your regular expression, it's what I normally use. I will go over a short intro to help get you started when creating a regular expression.

Match a single digit

`\d`

Match multiple digits

`\d+`

Match X amount of digits (if you want 3 digits; {3})

`\d{x}`

Match a single character

`\w`

Match any character or digit

`.`

Match any digit or character multiple times

`.+`

Match zero or more

`?`

So as an example, let's open the [cracked.to](#) website and view the source. We are going to use them as an example on how to scrape stuff. Let's scrape usernames and ID's.

Around line 2955 you will start to see the usernames

So copy that and paste it into the bottom textbox on [regex101](#). The top box is where we will begin to write our regex. Each entry is an anchor link which has the username and their ID.

Code `<a href="https://cracked.to/member.php?action=profile&uid=103683" title="17MOD1997">17MOD1997</a>`

In the top box put the following expression

```
Code member[.]php[?]action=profile&uid=(\d+)""  
title=""[a-zA-Z0-9&_\.-]+?"">([a-zA-Z0-9&_\.-]+?)</a>
```

Which should match that user (if they are logged in)

On the right in the image above you can see that we matched it. However, we only matched that one entry and we want them all. So now, let's replace the name and id with a regular expression.

```
Code member[.]php[?]action=profile&uid=(\d+)" title=".+?">(.*?)</a>
```

So now let's turn this into something we can use in C#. You will need to include the `RegularExpression` namespace (<https://docs.microsoft.com/en-us/dotnet/api/system.text.regularexpressions?view=netframework-4.7.2>). You can do that by adding the following line to the head of your code

```
Code using System.Text.RegularExpressions;
```

We are going to read a text file that has the source of the main cracked.to website, so we don't leave a footprint on their site while working with the regular expression demo. You can download the source [I am using here](https://pastr.io/raw/NehxTX) (<https://pastr.io/raw/NehxTX>). We can read the file using the `File.ReadAllLines()` method. We will need to add the `System.IO` namespace for this.

```
Code string src = File.ReadAllLines("testsrc");
```

So now let's apply our expression to the file we read above. Since the file we are using as a test is much different than how the source looks in your browser (in my case, Chrome) we are going to have to change our expression. Below is the image of the block of text on which we are going to use the expression.

Create a new Regex Object

```
Code Regex r = new
```

```
Regex(@"https://cracked[.]to/member[.]php[?]action=profile&uid=(\d+)&quot;  
title=&quot;([a-zA-Z0-9&_\.-]+)&quot;&gt;&lt;span");
```

Get the matches with MatchCollection

```
Code MatchCollection mc = r.Matches(src);
```

Now we can loop through the matches in the MatchCollection, showing the capture groups. The capture groups is the data in parenthesis in the regular expression.

```
Code foreach(Match m in mc)
```

```
{  
    Console.WriteLine("ID: " + m.Groups[1].Value + " \t Name: " + m.Groups[2].Value);  
}
```

Mr. Robot - April 15, 2019, 4:25 pm

When's my thread coming OP.

realsunjester - April 16, 2019, 6:51 am

(Apr 15, 2019 04:25 PM)Mr. Robot Wrote: [\[CITE\]](#) When's my thread coming OP.

what thread is that? lemme know

+Haze - April 16, 2019, 7:29 am

I understand you're trying to contribute to the forums, but this thread doesn't really have value. Not only do you misunderstand Regex and it's applications, but the examples you've shown are really bad.

1) In your example you used Regex to parse an HTML document, meanwhile, an HTML parsing library would be a lot better, not just because of performance, but because of maintainability. It's extremely difficult debugging fat regex expressions, and that's why you should them sparingly. What you have just shown can be done using AngleSharp parsing library and offer way more readability and maintainability than your regex expressions.

2) I've noticed that members on this forum seem to think that Regex is for parsing, but the truth is, it isn't. Regex is designed for pattern matching.

3)

Quote:Regular expressions is great for finding specific strings in a block of text, and it can be MUCH faster.

Nope, Regex is far slower than string functions. It's only in the past decade that Regex has been used more, and that's because machines are now more powerful, and can easily handle regex expressions. Back when regex was created in the 90s, they were very slow as computers then lacked power.

realsunjester - April 16, 2019, 7:35 am

(Apr 16, 2019 07:29 AM)+Haze Wrote: [\[CITE\]](#) I understand you're trying to contribute to the forums, but this thread doesn't really have value. Not only do you misunderstand Regex and it's applications, but the examples you've shown are really bad.

1) In your example you used Regex to parse an HTML document, meanwhile, an HTML parsing library would be a lot better, not just because of performance, but because of maintainability. It's extremely difficult debugging fat regex expressions, and that's why you should them sparingly. What you have just shown can be done using AngleSharp parsing library and offer way more readability and maintainability than your regex expressions.

2) I've noticed that members on this forum seem to think that Regex is for parsing, but the truth is, it isn't. Regex is designed for pattern matching.

3)

Nope, Regex is far slower than string functions. It's only in the past decade that Regex has been used more, and that's because machines are now more powerful, and can easily handle regex expressions. Back when regex was created in the 90s, they were very slow as computers then lacked power.

thanks for your opinion

vaiperboy - April 16, 2019, 9:43 am

using regex for parsing a html doc is by far the worst option

realsunjester - April 16, 2019, 9:59 am

im sorry if you are all confused but its a tutorial on using regular expressions, not parsing html. thanks you for your opinions though.

BahNahNah - April 16, 2019, 10:48 pm

(Apr 16, 2019 07:35 AM)realsunjester Wrote: [\[CITE\]](#) thanks for your opinion
Sorry boi but aint no opinion

<https://stackoverflow.com/a/1732454/2691759>

(Apr 15, 2019 11:24 AM)realsunjester Wrote: [\[CITE\]](#) You would normally use String.Split when there is only one occurrence of a string in the block your searching.

Im not sure you understand what String.Split is used for either. One occurrence of what?

Yes, regex is a fast-to-prototype, hacky way to extract very small specific data. But it is not a substitute for a parser.

(Apr 16, 2019 09:59 AM)realsunjester Wrote: [\[CITE\]](#) im sorry if you are all confused but its a tutorial on using regular expressions, not parsing html. thanks you for your opinions though.

If you want to make an educational post on how to use regex, you should probably use it correctly.

vaiperboy - April 17, 2019, 7:20 am

(Apr 15, 2019 11:24 AM)realsunjester Wrote: [\[CITE\]](#) Code foreach(Match m in mc)

```
{  
    Console.WriteLine("ID: " + m.Groups[1].Value + " \t Name: " + m.Groups[2].Value);  
}
```

please, i hate foreach loops

use

```
Code mc.ToList().ForEach(f => {Console.WriteLine($"ID: { m.Groups[1].Value } \t Name: {  
m.Groups[2].Value }");
```

no need for tolist i believe, idk if its enumerable or not

Owl - April 17, 2019, 9:11 am

(Apr 15, 2019 11:24 AM)realsunjester Wrote: [\[CITE\]](#) Match a single character

\w

Match any character or digit

.
lol

(Apr 16, 2019 09:59 AM)realsunjester Wrote: [\[CITE\]](#) im sorry if you are all confused but its a tutorial on using regular expressions, not parsing html.

All you are doing is using regex to parse html but thx for your opinion.

(Apr 17, 2019 07:20 AM)vaiperboy Wrote: [\[CITE\]](#) please, i hate foreach loops

use

```
Code mc.ToList().ForEach(f => {Console.WriteLine($"ID: { m.Groups[1].Value } \t Name: {  
m.Groups[2].Value }");
```

no need for tolist i believe, idk if its enumerable or not

Of course it's enumerable it's a MatchCollection??? The one thing OP got right and you call it out.

```
Code mc.OfType<Match>().ToList().ForEach(f =>
```

To make that nonsense work.

JMuller - April 17, 2019, 9:19 am

@OP, it's a good opportunity to update your knowledge, don't let your ego get in the way. These aren't opinions, they're widely accepted truths and best practices.

vaiperboy - April 17, 2019, 10:11 am

(Apr 17, 2019 09:11 AM)Owl Wrote: [\[CITE\]](#) Of course it's enumerable it's a MatchCollection??? The one thing OP got right and you call it out.well to be fair I didn't read the thread just got to the html part and ditched it

also i didnt sleep for 21 hours so far. give me a break

HFDone - April 17, 2019, 12:17 pm

(Apr 17, 2019 09:11 AM)Owl Wrote: [\[CITE\]](#) lol

All you are doing is using regex to parse html but thx for your opinion.

Of course it's enumerable it's a MatchCollection??? The one thing OP got right and you call it out.

Code mc.OfType<Match>().ToList().ForEach(f =>
To make that nonsense work.
linq is slower so its depends on what you using it for

SneakyHF - April 17, 2019, 12:27 pm

Thank you for this beautiful tutorial, I can't stress enough how useful this can be. Everyone needs to get on this yo

itcorp - April 18, 2019, 5:39 pm

(Apr 17, 2019 12:17 PM)HFDone Wrote: [\[CITE\]](#) linq is slower so its depends on what you using it for
Here's a small test of speed with good and bad examples of code

Code class Program

```
{  
    static void Main(string[] args)  
    {  
        var src = new WebClient().DownloadString("https://pastr.io/raw/NehxTX");  
        Regex r = new  
Regex(@"\"https:\\Vcracked[.]to\\member[.]php[?]*action=profile&uid=(\\d+)&quot;  
title=&quot;([a-zA-Z0-9&_\\.]+)&quot;&gt;&lt;span");  
        MatchCollection mc = r.Matches(src);  
  
        Test1(mc);  
  
        Test1a(mc);  
  
        Test2(mc);  
    }  
}
```

```

    Test3(mc);

    Test4(r,src);

    Console.ReadLine();
}

static void Test1(MatchCollection items)
{
    Stopwatch s = new Stopwatch();
    s.Start();

    foreach (Match item in items)
    {
        Console.WriteLine("ID: " + item.Groups[1].Value + " \t Name: " + item.Groups[2].Value);
    }

    Console.WriteLine("Test1");
    Console.WriteLine("Time:" + s.ElapsedMilliseconds);
    Console.WriteLine("=====");
}

static void Test1a(MatchCollection items)
{
    Stopwatch s = new Stopwatch();
    s.Start();

    foreach (var item in items.OfType<Match>())
    {
        Console.WriteLine("ID: " + item.Groups[1].Value + " \t Name: " + item.Groups[2].Value);
    }

    Console.WriteLine("Test1a");
    Console.WriteLine("Time:" + s.ElapsedMilliseconds);
    Console.WriteLine("=====");
}

static void Test2(MatchCollection items)
{
    Stopwatch s = new Stopwatch();
    s.Start();

    for (int i =0; i < items.Count; i++)
    {
        Console.WriteLine($"ID: {items[i].Groups[1].Value } \t Name: { items[i].Groups[2].Value
});
    }

    Console.WriteLine("Test2");
    Console.WriteLine("Time:" + s.ElapsedMilliseconds);
    Console.WriteLine("=====");
}

```

```

static void Test3(MatchCollection items)
{
    Stopwatch s = new Stopwatch();
    s.Start();

    items.OfType<Match>().ToList().ForEach( i =>

        Console.WriteLine($"ID: {i.Groups[1].Value } \t Name: { i.Groups[2].Value }")
    );

    Console.WriteLine("Test3");
    Console.WriteLine("Time:" + s.ElapsedMilliseconds);
    Console.WriteLine("=====");
}

static void Test4(Regex matches,string src)
{
    Stopwatch s = new Stopwatch();
    s.Start();

    matches.Matches(src).Cast<Match>().ToList().ForEach(i => Console.WriteLine($"ID:
{i.Groups[1].Value } \t Name: { i.Groups[2].Value }") );

    Console.WriteLine("Test4");
    Console.WriteLine("Time:" + s.ElapsedMilliseconds);
    Console.WriteLine("=====");
}
}

```

0xDEAD10CC - April 19, 2019, 10:08 am

(Apr 17, 2019 07:20 AM)vaiperboy Wrote: [\[CITE\]](#) please, i hate foreach loops

use
Code mc.ToList().ForEach(f => {Console.WriteLine(\$"ID: { m.Groups[1].Value } \t Name: { m.Groups[2].Value }");
no need for tolist i believe, idk if its enumerable or not
Yeah let's just create a list object for no reason so we can use its ForEach method. Must be more efficient, since it is a single line after all. ? You also don't seem to realize that ForEach() is not a LINQ method, it's specifically for the List<T> class.

The relevant parts of the Regex class nobody talked about in this entire thread either with compiled regex, and others. And the definitions are wrong:

(Apr 15, 2019 11:24 AM)realsunjester Wrote: [\[CITE\]](#) Match a single digit
\d

Match multiple digits
\d+

Match X amount of digits (if you want 3 digits; {3})
\d{x}

Match a single character

\w

Match any character or digit

.

Match any digit or character multiple times

.*

Match zero or more

?

? doesn't match zero or more*, that's what the * does in regex.

? matches zero or one of the corresponding token.

As with ., the definition of what this represents in .NET changes with the flags you're using, which aren't even mentioned in this entire thread.

HFDone - April 21, 2019, 10:57 am

(Apr 18, 2019 05:39 PM)itcorp Wrote: [\[CITE\]](#) Here's a small test of speed with good and bad examples of code

...

using Console.WriteLine in speed test is really a BAD idea

+Haze - April 21, 2019, 11:34 am

(Apr 21, 2019 10:57 AM)HFDone Wrote: [\[CITE\]](#) using Console.WriteLine in speed test is really a BAD idea

Theres nothing wrong with using System.WriteLine it's just the way it's used. In his example it basically produces an inaccurate result. A solution would be to time it inside the foreach block and add up the times or something.

itcorp - April 21, 2019, 1:18 pm

(Apr 21, 2019 11:34 AM)+Haze Wrote: [\[CITE\]](#) Theres nothing wrong with using System.WriteLine it's just the way it's used. In his example it basically produces an inaccurate result. A solution would be to time it inside the foreach block and add up the times or something.

Thanks :P

(Apr 19, 2019 10:08 AM)0xDEAD10CC Wrote: [\[CITE\]](#) Yeah let's just create a list object for no reason so we can use its ForEach method. Must be more efficient, since it is a single line after all. ? You also don't seem to realize that ForEach() is not a LINQ method, it's specifically for the List<T> class.

The relevant parts of the Regex class nobody talked about in this entire thread either with compiled regex, and others. And the definitions are wrong:

? doesn't match zero or more*, that's what the * does in regex.

? matches zero or one of the corresponding token.

As with ., the definition of what this represents in .NET changes with the flags you're using, which aren't even mentioned in this entire thread.

I saw you online and I did not want to interfere,I was expecting your to comment on this...And of course, as always precise observation.Respect!!

Joel - April 23, 2019, 8:03 pm

(Apr 16, 2019 07:29 AM)+Haze Wrote: [\[CITE\]](#) I understand you're trying to contribute to the forums, but this thread doesn't really have value. Not only do you misunderstand Regex and it's applications, but the examples you've shown are really bad.

1) In your example you used Regex to parse an HTML document, meanwhile, an HTML parsing library would be a lot better, not just because of performance, but because of maintainability. It's extremely difficult debugging fat regex expressions, and that's why you should use them sparingly. What you have just shown can be done using AngleSharp parsing library and offer way more readability and maintainability than your regex expressions.

2) I've noticed that members on this forum seem to think that Regex is for parsing, but the truth is, it isn't. Regex is designed for pattern matching.

3)

Nope, Regex is far slower than string functions. It's only in the past decade that Regex has been used more, and that's because machines are now more powerful, and can easily handle regex expressions. Back when regex was created in the 90s, they were very slow as computers then lacked power.

No offence but fuck off lmao.

Edit: This thread wasn't made to show you how to parse HTML, it's made to show you the basics of RegEx. Why the fuck are you all so stupid. For people with a superiority complex you all lack basic comprehension skills.

BahNahNah - April 24, 2019, 12:19 am

(Apr 23, 2019 08:03 PM)Joel Wrote: [\[CITE\]](#) No offence but fuck off lmao.

Edit: This thread wasn't made to show you how to parse HTML, it's made to show you the basics of RegEx. Why the fuck are you all so stupid. For people with a superiority complex you all lack basic comprehension skills.

Should a thread about regex not demonstrate how to use regex properly? Or maybe i missed the point of the thread.

Be patient, im autistic.

Joel - April 24, 2019, 5:58 am

(Apr 24, 2019 12:19 AM)BahNahNah Wrote: [\[CITE\]](#) Should a thread about regex not demonstrate how to use regex properly? Or maybe i missed the point of the thread.

Be patient, im autistic.

It probably should. But that isn't what's being done here so why judge it on it?

If he opened the thread with "Hey I'm going to teach you how to use Regular Expressions to parse HTML" then I'm all about the feedback. But he didn't so what's the point in trying to correct him when that's not even what he said.

Owl - April 24, 2019, 6:46 am

(Apr 24, 2019 05:58 AM)Joel Wrote: [\[CITE\]](#) It probably should. But that isn't what's being done here so why judge it on it?

If he opened the thread with "Hey I'm going to teach you how to use Regular Expressions to parse

HTML" then I'm all about the feedback. But he didn't so what's the point in trying to correct him when that's not even what he said.

Maybe not what he said but that's exactly what he did, this tutorial only demonstrates parsing from HTML.

What was mentioned about regex itself is wrong. The quantifier and class definitions are wrong (as I pointed out with the same effort as OP's replies), as is the claim regex is faster against string splitting. If I really wanna nitpick, he confused ReadAllLines with ReadAllText in his example as well. 0 proofreading.

0xDEAD10CC - April 24, 2019, 2:47 pm

(Apr 23, 2019 08:03 PM)Joel Wrote: [\[CITE\]](#) No offence but fuck off lmao.

Edit: This thread wasn't made to show you how to parse HTML, it's made to show you the basics of RegEx. Why the fuck are you all so stupid. For people with a superiority complex you all lack basic comprehension skills.

If you had any clue before you decided to make a post like this you could have observed how terribly inaccurate this thread even taught the basics of Regex in that context. Not sure why or what you are defending here. Did you actually attempt to read the first post or any of the replies? Or do you not have sufficient knowledge of Regex to determine whether this is a good thread or not? -- In which case you probably shouldn't be posting with an opinion like that because it isn't justified.

>> Regular expressions can be a very useful tool in your arsenal when creating hacking tools. Doesn't explain or go into detail about this.

>> You should use regular expressions when split just won't cut it. You would normally use String.Split when there is only one occurrence of a string in the block you're searching. Regular expressions are great for finding specific strings in a block of text, and it can be MUCH faster.

Regex isn't an alternative to string Split(), nor is it determinably "faster", not to mention in most cases it is slower -- if we disregard the fact that this is like comparing apples and oranges.

>> Match a single character \w

Not really accurate either, and the terminology doesn't even explain what is meant here properly. '9' ' ' etc are all characters too, yet that's not what \w matches.

... Which only makes the next definition extremely ambiguous:

>> Match any character or digit .

-- As well as any other mentions of the word "character" here because the definition is now contrived.

>> Match zero or more ?

This is actually supposed to mean something along the lines of "optional", not necessarily zero or "MORE" because it doesn't define that it can match any more than a single item in context.

Already out of the above, that's about 2/7 Regex elements properly defined. The rest of the thread is just as bad.

The point of this thread is kind of ambiguous as well. I can't tell whether he's trying to teach how to

program something specific, how to extract data, or how this ties into making a hacking tool as mentioned in the very beginning of the thread. With a poor example, incorrect definitions, bad terminology, and lack of completion for this to be specifically a Regex tutorial I can't say this thread is of much value.

Defend this thread all you want though.

itcorp - April 24, 2019, 3:56 pm

(Apr 24, 2019 03:40 PM)Joel Wrote: [\[CITE\]](#) Lol idec that much calm down.

Edit: Not reading anything you posted I aint got time for silly games.

I must say that I agree that we should not judge, and should contribute to the forum but I do not think it's about condemnation here but in directing. What 0xDEAD10CC brings up people is misunderstood but in many cases all he brought out is precisely and argued and I respect him for that. Do not get me wrong, all the best.

0xDEAD10CC - April 24, 2019, 4:07 pm

(Apr 24, 2019 03:40 PM)Joel Wrote: [\[CITE\]](#) Lol idec that much calm down.

Edit: Not reading anything you posted I aint got time for silly games.

The only games being played here, if that's what you take away from this, would be ones initiated by you.

>> Edit: This thread wasn't made to show you how to parse HTML, it's made to show you the basics of RegEx. Why the fuck are you all so stupid. For people with a superiority complex you all lack basic comprehension skills.

Explain how we are all "so stupid"? There are actually people here (not including you), who understand why this thread isn't a good piece of learning material for those who are actually trying to learn about Regex. Most of the points made throughout this thread that you have attempted to argue with, are completely valid and justified statements as well. I pointed some more things out for you too if you haven't been able to figure it out yet.

If nobody corrected any of the material that was posted on the internet how well do you think people would be able to learn from the information? There's valid reasons for constructive criticism, and you seem to be just too narrow minded to see the benefit of it. Would you be mad at your professors if they taught you all the wrong information and this led to you failing an exam in university? Nobody said contributions are not welcome here but you need to learn that feedback and criticism is part of the process.

I'm dishing out just plain facts, nothing personal against anyone; just using common sense. Maybe you're the one that should take a step back and learn to comprehend what others are saying? And if comprehension isn't your issue, maybe next time try actually reading the posts before firing at others.

Just a suggestion.

edit: Anyways, I already posted all of the feedback that I was able to provide, so there's nothing left for me to comment on within this thread now.

itcorp - April 24, 2019, 4:23 pm

(Apr 24, 2019 04:07 PM)0xDEAD10CC Wrote: [\[CITE\]](#) The only games being played here, if that's what you take away from this, would be ones initiated by you.

>> Edit: This thread wasn't made to show you how to parse HTML, its made to show you the basics of RegEx. Why the fuck are you all so stupid. For people with a superiority complex you all lack basic comprehension skills.

Explain how we are all "so stupid"? There are actually people here (not including you), who understand why this thread isn't a good piece of learning material for those who are actually trying to learn about Regex. Most of the points made throughout this thread that you have attempted to argue with, are completely valid and justified statements as well. I pointed some more things out for you too if you haven't been able to figure it out yet.

If nobody corrected any of the material that was posted on the internet how well do you think people would be able to learn from the information? There's valid reasons for constructive criticism, and you seem to be just too narrow minded to see the benefit of it. Would you be mad at your professors if they taught you all the wrong information and this led to you failing an exam in university?

I'm dishing out just plain facts, nothing personal against anyone; just using common sense. Maybe you're the one that should take a step back and learn to comprehend what others are saying? And if comprehension isn't your issue, maybe next time try actually reading the posts before firing at others.

Just a suggestion.

edit: Anyways, I already posted all of the feedback that I was able to provide, so there's nothing left for me to comment on within this thread now. As far as I'm concerned, I like how you answer, as I have indicated to you last time. If I could I'd give you more reputation and bytes because of that. Go on and I think that many can learn a lot from you the way you give answers.... Just wish your brain a few minutes.

realsunjester - April 30, 2019, 5:21 pm

for those who need more clarification on the specifics for C# regex

<https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>