Ruby master - Bug #13970

Base64 urlsafe_decode64 unsafe use of tr.

10/04/2017 01:29 PM - shanna (Shane Hanna)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>ruby -v:</td>
<td></td>
</tr>
<tr>
<td>Backport:</td>
<td>2.3: UNKNOWN, 2.4: UNKNOWN</td>
</tr>
</tbody>
</table>

**Description**

A lot of the base64 module lacks duck typing or nice errors.

For example the urlsafe_decode64 function never checks str is something that behaves like a string and will respond to tr. If you pass nil by mistake you end up with the dreaded "can't call method on (n)" rather than an informative error.

```ruby
def urlsafe_decode64(str)
    # NOTE: RFC 4648 does say nothing about unpadded input, but says that
    # "the excess pad characters MAY also be ignored", so it is inferred that
    # unpadded input is also acceptable.
    str = str.tr("-_", "+/")
    if !str.end_with?("=") && str.length % 4 != 0
        str = str.ljust((str.length + 3) & ~3, ")=")
    end
    strict_decode64(str)
end
```

Raising an error or silently failing if the argument doesn't respond to tr (or to_s.tr) both seem preferable to errors raised by the internal implementation but I'm wondering if there is a preferred approach in Rubys stdlib?

**History**

#1 - 08/25/2019 08:36 PM - jeremyevans0 (Jeremy Evans)
- Status changed from Open to Rejected

I don't think this is a bug. Most pure Ruby code assumes and does not check that method arguments respond to all methods that the code expects them to respond to. It's generally considered a smell to take code like:

```ruby
def foo(bar, baz)
    bar.x + baz.y
end
```

and change it to:

```ruby
def foo(bar, baz)
    raise ArgumentError, "bar does not respond to x" unless bar.respond_to?(:x)
    raise ArgumentError, "baz does not respond to y" unless baz.respond_to?(:y)
    bar.x + baz.y
end
```

Especially if bar or baz could be instances of subclasses of BasicObject and not Object with the x and y methods defined, respectively.