tjgerot External links and author attribution … Latest commit 274307f on 7 Nov 2016

..

Hamlet.docx	Added Shakespeare to angstromCTF (#1033)	a year ago
README.md	External links and author attribution	6 months ago
shakespeare-1.jpg	Added Shakespeare to angstromCTF (#1033)	a year ago
shakespeare-2.jpg	Added Shakespeare to angstromCTF (#1033)	a year ago

README.md

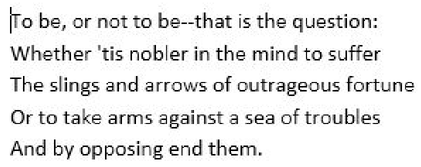
angstromCTF 2016 : shakespeare-60

Category: Crypto Points: Solves: Description:

> We have uncovered a Shakespearean-era transmission that seems perfectly ordinary. Can you help us find the hidden message in this Hamlet soliloquy? Hint: Who do some claim wrote the plays normally attributed to Shakespeare? The flag will be a lower-case string with no spaces.

Write-up

We are given a single Word Document for this challenge. Upon opening it with Word 2016 we find the opening of the famous soliloquy from Shakespeare's Hamlet.

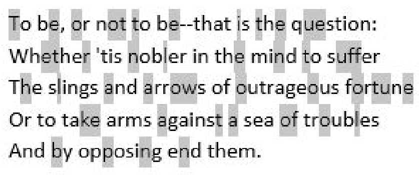

To be, or not to be--that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles
And by opposing end them.

There's nothing obvious that stands out in the text, and there doesn't seem to be anything embedded in the document. Following the hint, I searched for "Shakespeare conspiracies", and the third result gave me what I was looking for: The Shakespeare Authorship Question

To quote the page: "The Shakespeare authorship question is the argument that someone other than William Shakespeare of Stratford-upon-Avon wrote the works attributed to him"..."The controversy has since spawned a vast body of literature, and more than 80 authorship candidates have been proposed, the most popular being Sir Francis Bacon; Edward de Vere, 17th Earl of Oxford; Christopher Marlowe; and William Stanley, 6th Earl of Derby."

Reading these names prompted me to start searching for encryption methods related to the supposed authors. Sir Francis Bacon was first, so I started off by searching for "Bacon encoding"... And what do you know! It turns out Sir Bacon devised a form of steganography that is commonly called the Baconian Cipher!

The Wikipedia article does a pretty good job of explaining it, so I won't go into the details here. To summarise, Baconian Ciphers encode data in the presentation of the text, and not the text itself. After a bit of thought and experimentation, I realised that the body of text we had been given used two different (but visibly identical) fonts: Calibri Light (Headings) and Calibri (Body).


To be, or not to be--that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles
And by opposing end them.

I used Word's "Select Text with Similar Formatting" tool to grab all of one of the fonts, then used Shift+F3 to make the selection all uppercase. I also removed any non-alphabetic from the string at this point, which left me with: tOBeoRNottOBeTHATiStHeQueSTIONWHetHErTISnOBlERINTHeMInDtOSuffERThESlINGSaNDaRRowSOFoUTRAGEoUSFoRtuNeORTOTAkeARMSaGAINStaSEAofTRoUBlESANDbYOPpOSInGEnDTHEM

This then needed to be converted to the A/B string that Baconian Ciphers use. I did this with a couple of lines of Python.

```
intext = "tOBeoRNottOBeTHATiStHeQueSTIONWHetHErTISnOBlERINTHeMInDtOSuffERThESlINGSaNDaRRowSOFoUTRAGEoUSFoR
outtext = ""
for c in intext:
    if c.isupper():
            outtext += "B"
    else:
            outtext += "A"

print outtext
'ABBAABBAAABBABBBBABABABAABBBBBBBAABBABBBABBABBBBBBABBABABBAAABBBABBABBBBABBABBAABBBABBBBBBABBBABAABABBBBB
```

Finally, I plugged the resulting string into an online Baconian Cipher Decoder and fiddling around with the parameters I get this:

> theflagisastreetcarnameddeqire

I must have gotten something very slightly wrong somewhere down the line... Anyway, the flag is easy to work out from here.

Flag: *astreetcarnameddesire*

Other write-ups and resources

- Jashan Bhorra