Challenge in Web App Exploitation – Command Injection - NodeJS. Can be put in Easy category.   
  
***Challenge:***   
  
Step into a world where the lines between reality and the digital realm blur, inspired by the brilliance of a woman who wove code like magic spells. In this web challenge, you find yourself in the midst of a digital tapestry spun by the enigmatic CipherWitch, drawing inspiration from the pioneering spirit of Natalie Silvanovich.

CipherWitch, an enigmatic figure inspired by Natalie Silvanovich’s legacy, has woven a complex Node.js web application. Uncover the hidden secrets that lie within its threads and discover the mysterious flag that she left behind. Dive deep into the heart of the code and bring forth the concealed truth.

Are you ready to embark on a journey of discovery, where the boundaries between reality and code become indistinguishable? Unravel the unseen threads of Node and reveal the hidden flag that pays tribute to the legacy of an extraordinary woman.  
  
Challenge description can be modified as per your POV.   
  
***Solution:***  
  
This is the solution for the NodeJS web application exploitation challenge. I was busy this week, so couldn’t write the solution in a great way, I had to fasten up.   
  
I created this website with NodeJS as its backend. The js file is programmed to create a child process for the query request. So, we can take this as a vulnerability and can gain RCE to the machine.   
  
The website is hosted on port 3000.   
  
A screenshot of a computer

Description automatically generated  
  
So, it’s basically a Domain to IP conversion thing. The example is given below to try!  
  
***http://localhost:3000/dns?ip=google.com***  
  
A screenshot of a computer

Description automatically generated  
And it gives us the output for any domain asked.   
  
Seems like a good website, but it has a vulnerability.   
  
The child\_process module allows executing system commands and scripts from a Node.js application. Functions like child\_process.exec() and child\_process.execSync() take a command string and execute it on the server. If user input is passed directly to these functions, it can lead to command injection.  
  
What if we try adding commands next to the URL?   
  
Like, [***http://localhost:3000/dns?ip=google.com;ls***](http://localhost:3000/dns?ip=google.com;ls)

A screenshot of a computer

Description automatically generated

We can see there a flag.txt file. Let’s try opening it.   
  
[***http://localhost:3000/dns?ip=google.com;cat flag.txt***](http://localhost:3000/dns?ip=google.com;cat%20flag.txts)  
  
A screenshot of a computer

Description automatically generated  
  
You can’t view it, as it has some filters!  
  
Even though you try accessing using “head”, “tail”, “more”, “less”, “awk”. It doesn’t bring you the flag.   
  
Let’s try using “grep” or a simple python code!  
  
***http://localhost:3000/dns?ip=google.com;%20grep%20%22chctf%22%20flag.txt***  
  
A screenshot of a computer

Description automatically generated  
  
This gives you the flag!   
  
Also, you can use Perl, or Python to read the contents of the flag.txt file.  
  
***http://localhost:3000/dns?ip=google.com;%20python%20c%20%27print(open(%22flag.txt%22).read())%27***  
  
A screenshot of a computer

Description automatically generated  
  
  
  
FLAG: ***chctf{1\_l0v3\_5p4wn1n6\_n0d3\_ch1ld\_pr0c355}***