Universidad Veracruzana

Licenciatura en Ingeniería de Software

Facultad de Estadística e Informática

CWE-209: Generation of Error Message

Containing Sensitive Information

Programación Segura (12800)

Guillermo Humberto Vera Amaro

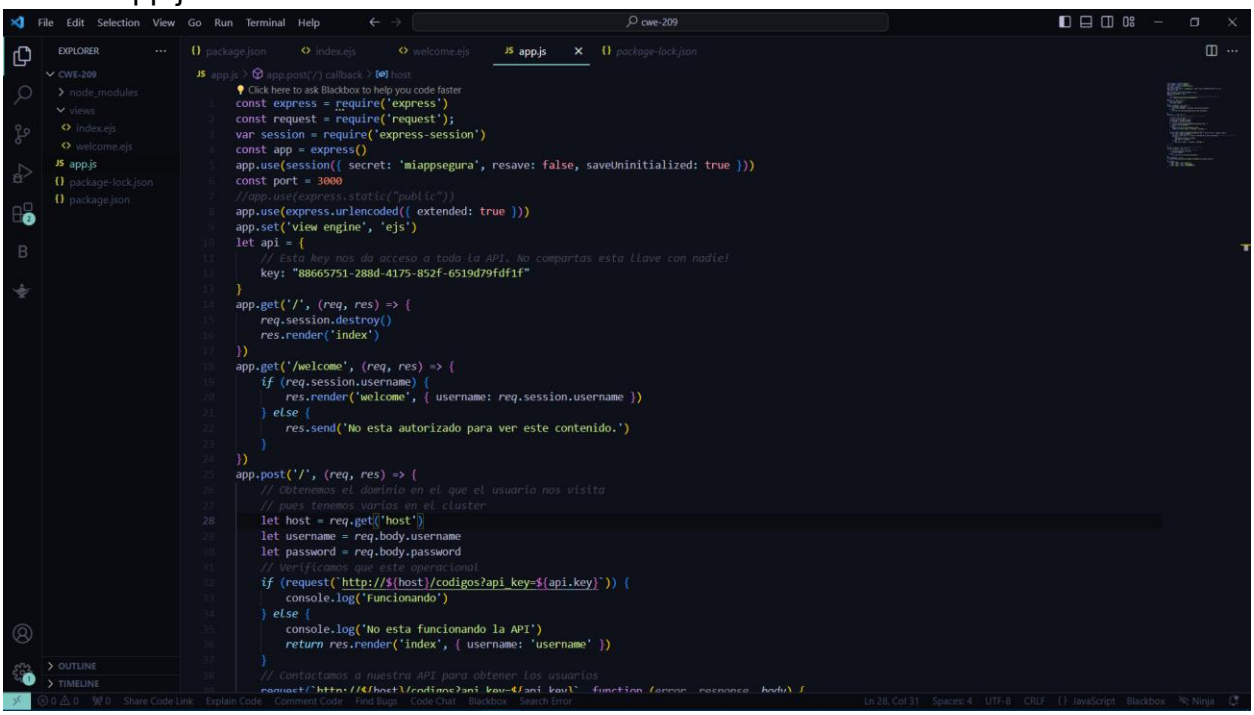
Daniel Mongeote Tlachy

Xalapa, Ver; martes 02 de abril del 2024

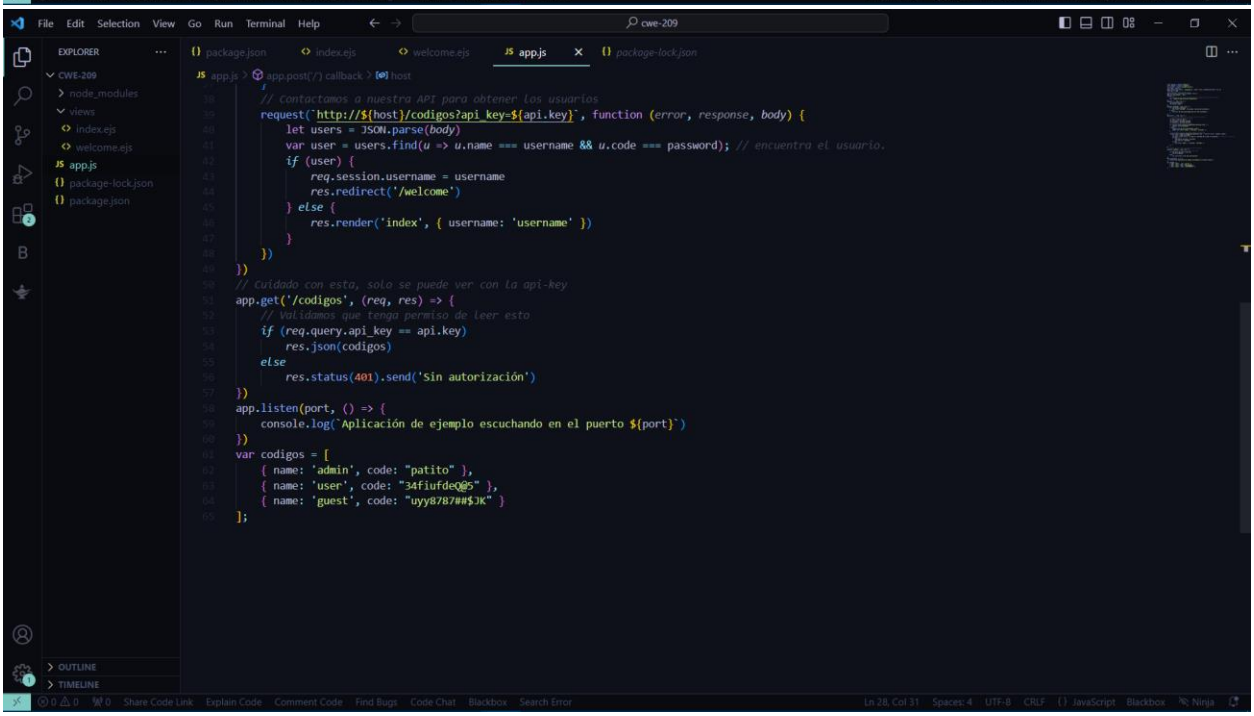
Instrucción. Elaborar una aplicación con Node.js que permita ejemplificar la vulnerabilidad Desreferencia de puntero NULL en una aplicación web cliente-servidor.

Coloca las siguientes capturas de pantalla, cada una con una descripción textual arriba de la captura.

a. Coloca la captura de pantalla de tu código fuente en Visual Studio Code del archivo app.js.

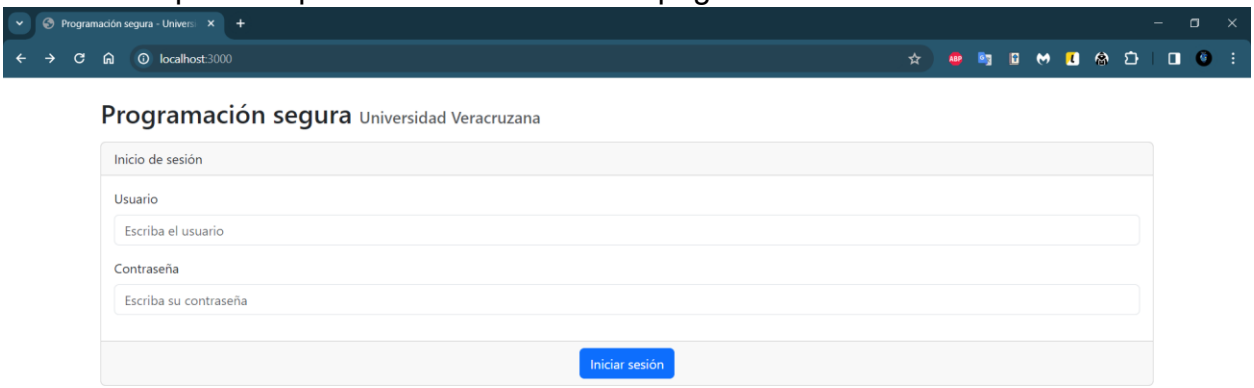


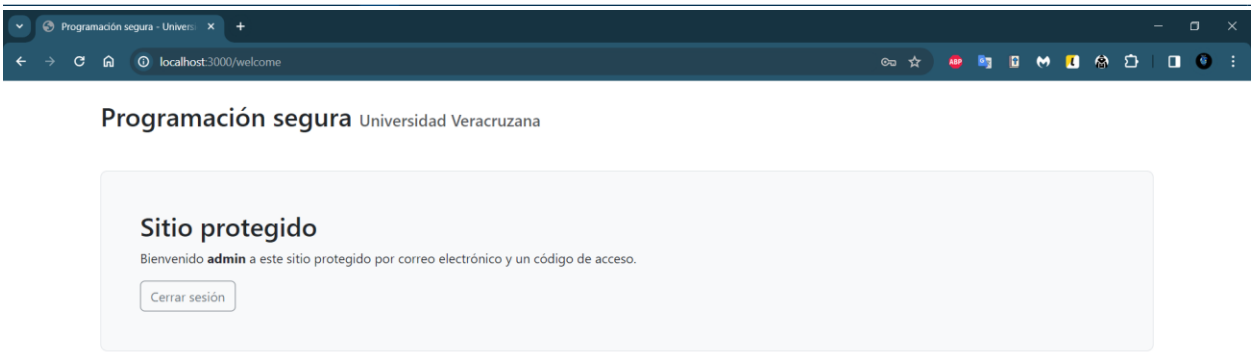
```
Click here to ask Blackbox to help you code faster
const express = require('express')
const request = require('request');
var session = require('express-session')
const app = express()
app.use(session({ secret: 'miappsegura', resave: false, saveUninitialized: true }))
const port = 3000
//app.use(express.static("public"))
app.use(express.urlencoded({ extended: true }))
app.set('view engine', 'ejs')
let api = {
    // Esta key nos da acceso a toda la API. No compartas esta llave con nadie!
    key: "88665751-288d-4175-852f-6519d79fdf1f"
}
app.get('/', (req, res) => {
    req.session.destroy()
    res.render('index')
})
app.get('/welcome', (req, res) => {
    if (req.session.username) {
        res.render('welcome', { username: req.session.username })
    } else {
        res.send('No esta autorizado para ver este contenido.')
    }
})
app.post('/', (req, res) => {
    // Obtenemos el dominio en el que el usuario nos visita
    // pues tenemos varios en el cluster
    let host = req.get('host')
    let username = req.body.username
    let password = req.body.password
    // Verificamos que este operacional
    if (request(`http://${host}/codigos?api_key=${api.key}`)) {
        console.log('Funcionando')
    } else {
        console.log('No esta funcionando la API')
        return res.render('index', { username: 'username' })
    }
    // Contactamos a nuestra API para obtener los usuarios
```

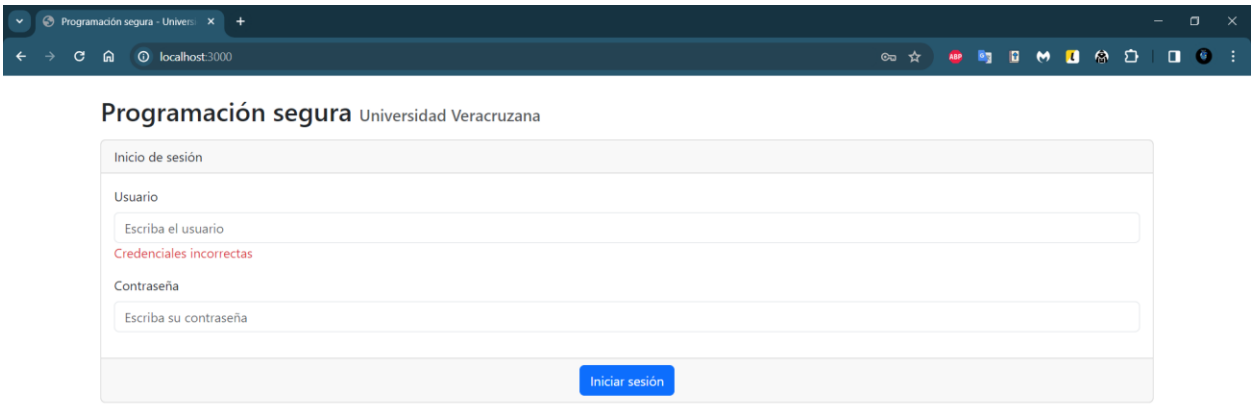


```
    // Contactamos a nuestra API para obtener los usuarios
    request(`http://${host}/codigos?api_key=${api.key}`, function (error, response, body) {
        let users = JSON.parse(body)
        var user = users.find(u => u.name === username && u.code === password); // encuentra el usuario.
        if (user) {
            req.session.username = username
            res.redirect('/welcome')
        } else {
            res.render('index', { username: 'username' })
        }
    })
})
// Cuidado con esta, solo se puede ver con la api-key
app.get('/codigos', (req, res) => {
    // Validamos que tenga permiso de leer esta
    if (req.query.api_key == api.key)
        res.json(codigos)
    else
        res.status(401).send('Sin autorización')
})
app.listen(port, () => {
    console.log(`Aplicación de ejemplo escuchando en el puerto ${port}`)
})
var codigos = [
    { name: 'admin', code: "patito" },
    { name: 'user', code: "34fiufdeQ@5" },
    { name: 'guest', code: "uyy8787##$JK" }
];
```

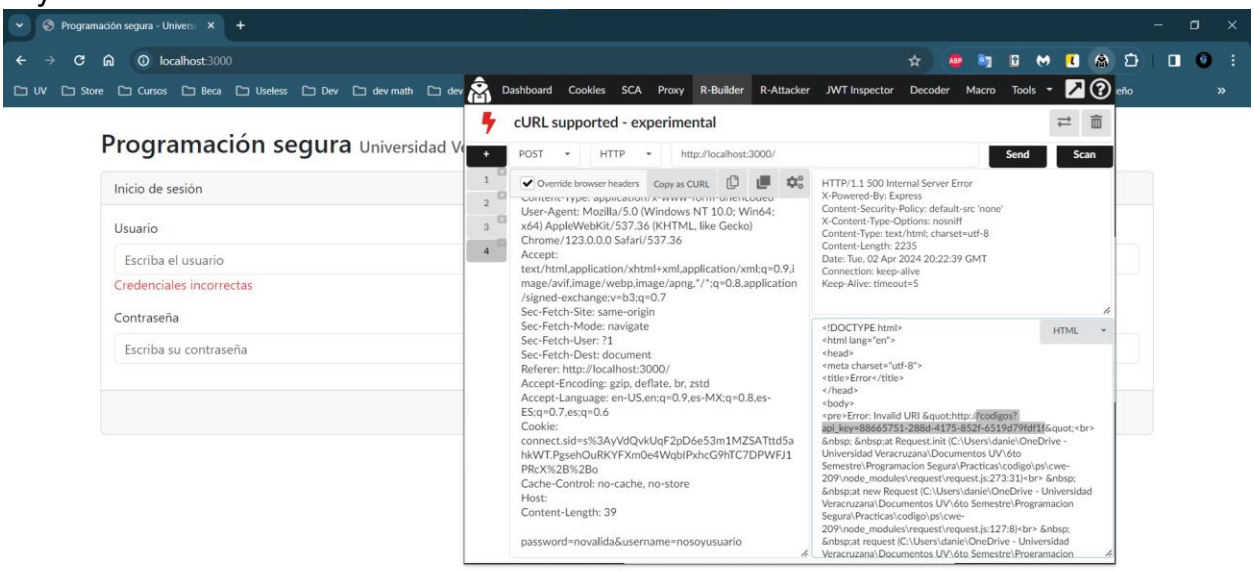
b. Coloca la captura de pantalla de la salida de la página index.html de autenticación.

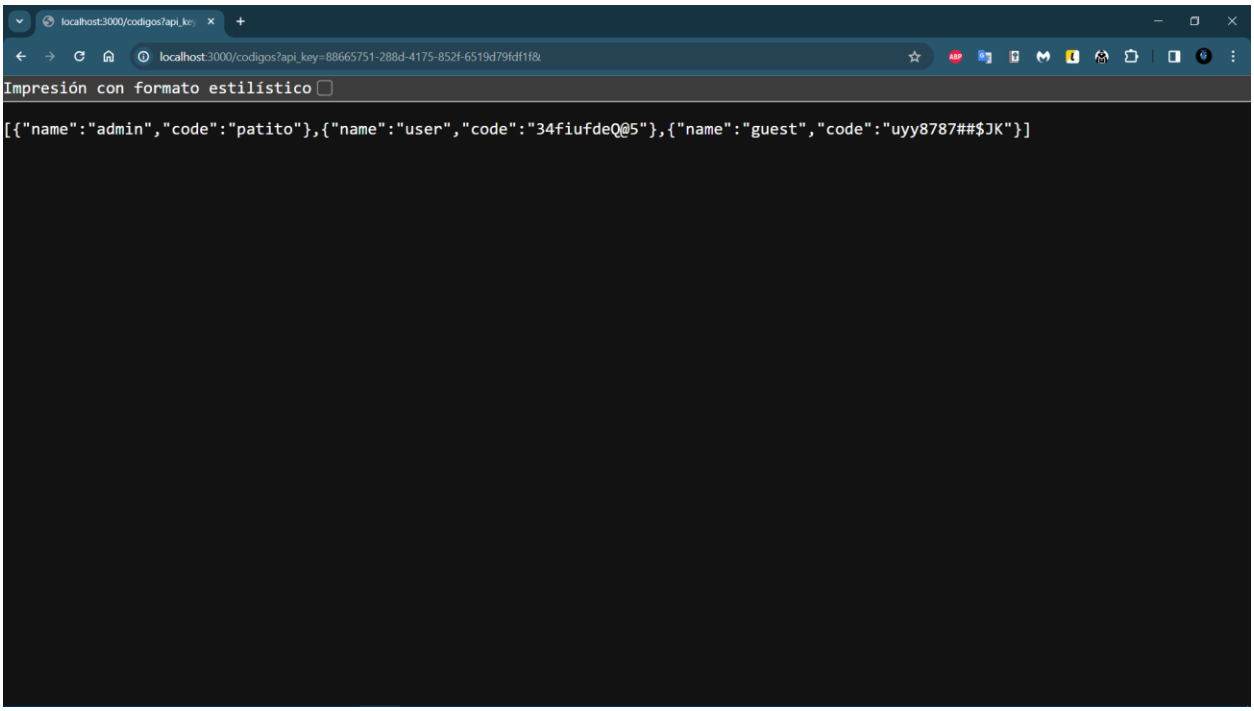






c. Coloca la captura de pantalla de la salida del error de Node.js donde se vea la api key revelada al atacante.





d. Publica tu código fuente en un proyecto público de GitHub. Coloca la URL del código fuente publicado en GitHub.
URL: https://github.com/danieltlachy/CWE-209-Generation-of-Error-Message-Containing-Sensitive-Information.git