Exercise (Instructions): Using Cookies

Objectives and Outcomes

In this exercise you will examine the use of cookies for authentication. The server will send a signed cookie to the client upon successful authentication, and expects the client to include the cookie with every subsequent request. At the end of this exercise, you will be able to:

- Set up your Express application to send signed cookies.
- Set up your Express application to parse the cookies in the header of the incoming request messages

Using cookie-parser

- Install the *cookie-parser* Express middleware in your basic-auth folder by typing the following at the prompt:

```
     npm install cookie-parser --save
```

- Create a new file named server-2.js and add the following code to it:

```
var express = require('express');
var morgan = require('morgan');
var cookieParser = require('cookie-parser');

var hostname = 'localhost';
var port = 3000;

var app = express();

app.use(morgan('dev'));

app.use(cookieParser('12345-67890-09876-54321')); // secret key

function auth (req, res, next) {

    if (!req.signedCookies.user) {
        var authHeader = req.headers.authorization;
        if (!authHeader) {
            var err = new Error('You are not authenticated!');
            err.status = 401;
            next(err);
            return;
        }
        var auth = new Buffer(authHeader.split(' ')[1], 'base64').toString().split(':');
        var user = auth[0];
        var pass = auth[1];
        if (user == 'admin' && pass == 'password') {
            res.cookie('user','admin',{signed: true});
            next(); // authorized
        } else {
            var err = new Error('You are not authenticated!');
            err.status = 401;
            next(err);
        }
    }
    else {
        if (req.signedCookies.user === 'admin') {
            next();
        }
        else {
            var err = new Error('You are not authenticated!');
            err.status = 401;
            next(err);
        }
    }
};

app.use(auth);

app.use(express.static(__dirname + '/public'));

app.use(function(err,req,res,next) {

            res.writeHead(err.status || 500, {
            'WWW-Authenticate': 'Basic',
            'Content-Type': 'text/plain'
        });
        res.end(err.message);
});


app.listen(port, hostname, function(){
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

- Save the changes, run the server and test the behavior.

Conclusions

In this exercise you examined the use of cookies for tracking authenticated users so that subsequent access to the server can be enabled without need for authentication.