

Vulnerability test on websites

1. Broken Access Control

Unprotected admin panel

The image shows a web browser window. The address bar displays the URL: `0a17002c044b1157812e3ee800d9003e.web-security-academy.net/robots.txt`. Below the address bar, the text `User-agent: *` and `Disallow: /administrator-panel` is visible. The main content area shows the Web Security Academy logo and the title "Unprotected admin functionality". A green button labeled "LAB" and a status "Not solved" are present. Below this, there is a section titled "Users" with a list of users: "wioner - Delete" and "carlos - Delete". At the bottom, a message "User deleted successfully!" is displayed, followed by another "Users" section showing "wiener - Delete". Navigation links for "Home" and "My account" are visible in the top right corner.

Once the administrator link is found, there is no access control which makes the application vulnerable.

2. Cryptographic failure


Information disclosure in error messages

Web Security Academy | Information disclosure in error messages

Submit solution | Back to lab description >>

Lightbulb Moments

★★★★☆ \$81.57



Internal Server Error: java.lang.NumberFormatException: For input string: "test"

```
at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
at java.base/java.lang.Integer.parseInt(Integer.java:668)
at java.base/java.lang.Integer.parseInt(Integer.java:786)
at lab.t.x.r.z.N(Unknown Source)
at lab.k.i.u.a.O(Unknown Source)
at lab.k.i.s.i.b.B(Unknown Source)
at lab.k.i.s.k.lambda$handleSubRequest$0(Unknown Source)
at c.z.i.a.lambda$null$3(Unknown Source)
at c.z.i.a.x(Unknown Source)
at c.z.i.a.lambda$uncheckedFunction$4(Unknown Source)
at java.base/java.util.Optional.map(Optional.java:260)
at lab.k.i.s.k.N(Unknown Source)
at lab.server.o.g.w.c(Unknown Source)
at lab.k.i.n.T(Unknown Source)
at lab.k.i.n.c(Unknown Source)
at lab.server.o.g.j.v.A(Unknown Source)
at lab.server.o.g.j.f.lambda$handle$0(Unknown Source)
at lab.t.u.n.y.c(Unknown Source)
at lab.server.o.g.j.f.B(Unknown Source)
at lab.server.o.g.c.x(Unknown Source)
at c.z.i.a.lambda$null$3(Unknown Source)
at c.z.i.a.x(Unknown Source)
at c.z.i.a.lambda$uncheckedFunction$4(Unknown Source)
at lab.server.zl.O(Unknown Source)
at lab.server.o.g.c.i(Unknown Source)
at lab.server.o.f.q.l(Unknown Source)
at lab.server.o.d.o(Unknown Source)
at lab.server.o.v.o(Unknown Source)
at lab.server.z_.P(Unknown Source)
at lab.server.z_.f(Unknown Source)
at lab.r.k.lambda$consume$0(Unknown Source)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
at java.base/java.lang.Thread.run(Thread.java:833)
```

Apache Struts 2 2.3.31

Version number of the framework is visible in the error message and is not encrypted.

3. Injection

Email: [xxx@xxx.xxx](#)

Password: xxx') OR 1 = 1 —]

```
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' OR 1 = 1 LIMIT 1 — ' ]
AND password = md5('1234');
```

- **xxx@xxx.xxx** ends with a single quote which completes the string quote
- **OR 1 = 1 LIMIT 1** is a condition that will always be true and limits the returned results to only one record.
- **— ' AND ...** is a SQL comment that eliminates the password part.

The image shows two screenshots of a web application. The top screenshot is the login page, titled "Login | Personal Contacts Manager v1.0". It has a form with "Email*" and "Password*" fields. The email field contains "xxx@xxx.xxx" and the password field contains "xxx') OR 1 = 1 —]". There is a "Remember me" checkbox and a "Submit" button. The bottom screenshot is the dashboard page, titled "Dashboard | Personal Contacts Manager v1.0". It has an "Add New Contact" button and a "Log Out" button. Below these is a table with 6 columns: ID, First Name, Last Name, Mobile No, Email, and Actions. The table contains 9 rows of contact data. The first row is highlighted in blue. The last row is also highlighted in blue. Below the table is a "Total Records Count: 9" label.

Dashboard | Personal Contacts Manager v1.0

Add New Contact Log Out

ID	First Name	Last Name	Mobile No	Email	Actions
1	mynams	jenefry	9898989898	admin@gmail.com	
67840	http://124.29.206.118:9999/trasur/	maiden	05465488498	djahhlsd2@gmail.com	Edit
67841		wgr	0565	dcaskj@gmail.com	Edit
67842	david	david	02167357	grush346@gmail.com	Edit
67843	david	david	02167357	grush346@gmail.com	Edit
67844	damian	dami	0217218721	grush346@gmail.com	Edit
67845	imam	uhuy	762873617831	grush346@gmail.com	Edit
67846	dima	Антоненко	+79516240323	tatarnikovafialka@mail.ru	Edit
67847	dima	Антоненко	+79516240323	tatarnikovafialka@mail.ru	Edit

Total Records Count: 9

4. Insecure Design

```
1 [
2   {
3     "name": "Electrician Ellie",
4     "role": "grunt"
5   },
6   {
7     "name": "Plumber Polly",
8     "role": "grunt"
9   },
10  {
11    "name": "Developer Friend",
12    "role": "developer"
13  },
14  {
15    "name": "You",
16    "role": "castle_leader"
17  }
18 ]
```

```
1 [
2   {
3     "name": "Electrician Ellie",
4     "role": "grunt"
5   },
6   {
7     "name": "Plumber Polly",
8     "role": "grunt"
9   },
10  {
11    "name": "Developer Friend",
12    "role": "developer"
13  },
14  {
15    "name": "Sir Snake Oil",
16    "role": "developer"
17  }
18 ]
```

```
import json

f = open('permissions.json')
employee_roles = json.load(f)

def remove_employee(employee_id, requesting_person):
    # Fetch role from permissions.json where name equals $requesting_person
    role = list(filter(lambda x:x["name"]==requesting_person, employee_roles))[0]['role']

    # Only allow the roles 'developer' and 'castle_leader' to delete employees
    if role == "developer" or role == "castle_leader":
        print(requesting_person, employee_id)
        cursor.execute("DELETE FROM employees WHERE ID = ?", [employee_id])
```

In the above case, the newly entered developer friend can delete the leader and increase his privilege because of the error in the design.

Here, the privilege of the can be controlled by:

```
[
  {
    "name": "Electrician Ellie",
    "role": ["receive_order", "submit_timecard"]
  },
  {
    "name": "Plumber Polly",
    "role": ["receive_order", "submit_timecard"]
  },
  {
    "name": "Developer Friend",
```

```

    "role": ["receive_order", "submit_timecard", "change_permissions", "edit_code",
"add_employee", "remove_employee", ...]
  },
  {
    "name": "Sir Snake Oil",
    "role": ["receive_order", "submit_timecard", "edit_code", ...]
  }
  {
    "name": "You",
    "role": [ ... ]
  }
]

```

5. Security Misconfiguration

First, Launch Webgoat and navigate to insecure configuration section



We can try out as many options as we can think of. All we need to find the URL of config file and we know that the developers follow kind of naming convention for config files. It can be anything that is listed below. It is usually done by BRUTE force technique.

- web.config
- config
- appname.config
- conf

