

Ben Gamliel

# Pentration test report

## AltoroJ

18/05/22

---

### Table of content

- **FINDING OVERVIEW 2**
  - **SEVERITY SCALE 2**
  - **METHODOLOGY 3**
  - **INFORMATION GATHERING 3**
  - **ENUMERATION 5**
  - **VULNERABILITY ASSESSMENT 6**
  - **CONCLUSION 17**
-

---

## FINDING OVERVIEW

The web application runs in a safe environment on a local VM inside a container, while conducting the penetration test, there were several critical vulnerabilities discovered with the Altoroj web application, I was able to expose numerous vulnerabilities that could be leveraged into different attack vectors such as XSS, SQL Injection, DDoS, ReDDoS, and many more

This was possible since the web application was written with consideration of app functionality but not app security most of the app was written in Java and Javascript.

## SEVERITY SCALE

**CRITICAL Severity Issue:** Poses an immediate danger to systems, networks, and/or data security and should be addressed as soon as possible. Exploitation requires little to no special knowledge of the target. Exploitation doesn't require highly advanced skills, training, or tools.

**HIGH Severity Issue:** Poses a significant danger to systems, networks, and/or data security. Exploitation commonly requires some advanced knowledge, training, skill, and/or tools. Issue(s) should be addressed promptly.

**MEDIUM Severity Issue:** Vulnerabilities should be addressed promptly. Exploitation is usually more difficult to achieve and requires special knowledge or access. Exploitation may also need social engineering as well as special conditions.

**LOW Severity Issue:** The danger of exploitation is unlikely as vulnerabilities offer little to no opportunity to compromise the system, network, and/or data security. It can be handled as time permits.

**INFORMATIONAL Issue:** Meant to increase the client's knowledge. Likely no actual threat.

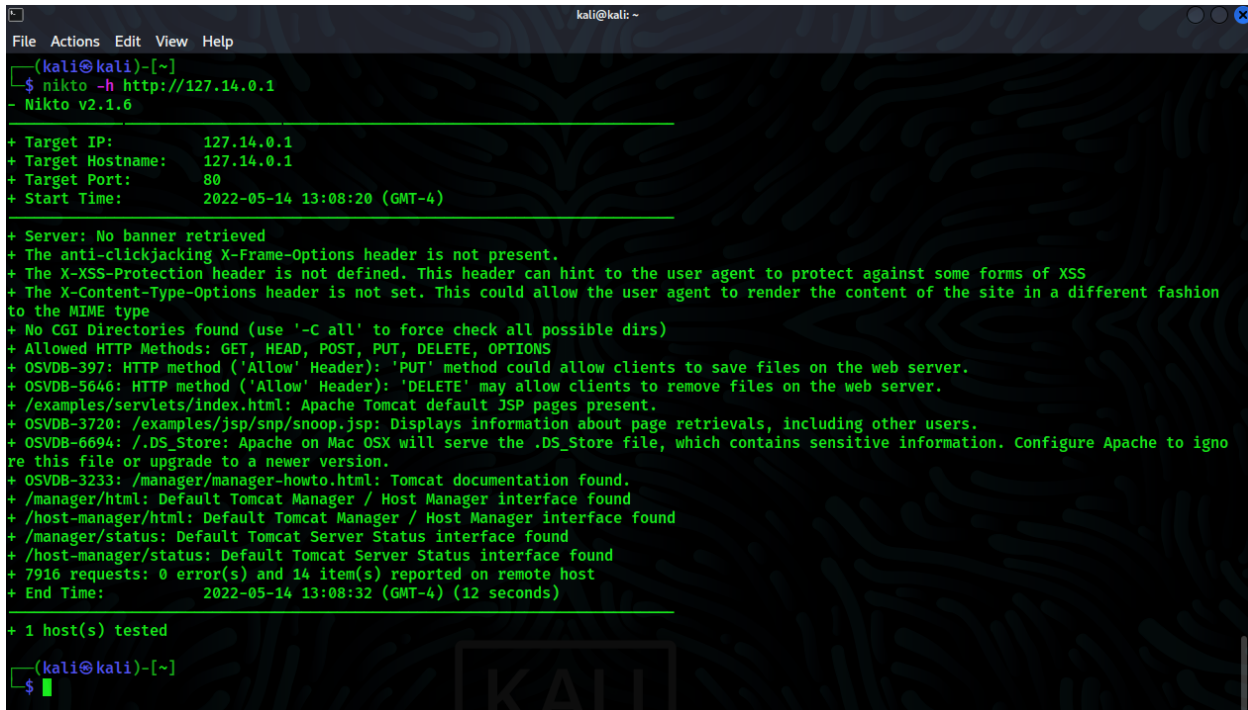
---

## METHODOLOGY

I used testing methods that are widely adopted and explained in the cyber-security assessment industry, which includes 5 phases: **Information Gathering, Enumeration, Vulnerability Assessment, Exploitation, and Reporting/Mitigation.**

## INFORMATION GATHERING

I was given the web application git repo and a public server, after setting up the running environment I started communicating with the predefined IP address with burp suite for my manual fuzzing, I also used some automated fuzzing tools installed on kali including Nmap, Nikto, wapiti, free web scanners such as observatory and the SAST tool nodejsscan for the javascript code on the server



```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nikto -h http://127.14.0.1
- Nikto v2.1.6

+ Target IP: 127.14.0.1
+ Target Hostname: 127.14.0.1
+ Target Port: 80
+ Start Time: 2022-05-14 13:08:20 (GMT-4)

+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ /examples/servlets/index.html: Apache Tomcat default JSP pages present.
+ OSVDB-3720: /examples/jsp/snp/snoop.jsp: Displays information about page retrievals, including other users.
+ OSVDB-6694: /.DS_Store: Apache on Mac OSX will serve the .DS_Store file, which contains sensitive information. Configure Apache to ignore this file or upgrade to a newer version.
+ OSVDB-3233: /manager/manager-howto.html: Tomcat documentation found.
+ /manager/html: Default Tomcat Manager / Host Manager interface found
+ /host-manager/html: Default Tomcat Manager / Host Manager interface found
+ /manager/status: Default Tomcat Server Status interface found
+ /host-manager/status: Default Tomcat Server Status interface found
+ 7916 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time: 2022-05-14 13:08:32 (GMT-4) (12 seconds)

+ 1 host(s) tested

(kali@kali)-[~]
$
```

## Wapiti scan report

## Raw Headers

HTTP/1.1	200 OK
Server	Apache-Coyote/1.1
Set-Cookie	JSESSIONID=88CAAD54C8DD1AEE614363CEC022FCC9; Path=/; HttpOnly
Content-Type	text/html; charset=ISO-8859-1
Transfer-Encoding	chunked
Date	Thu, 12 May 2022 20:32:32 GMT
Connection	close

## Missing Headers

Content-Security-Policy	<a href="#">Content Security Policy</a> is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets.
X-Frame-Options	<a href="#">X-Frame-Options</a> tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".
X-Content-Type-Options	<a href="#">X-Content-Type-Options</a> stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".
Referrer-Policy	<a href="#">Referrer Policy</a> is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.
Permissions-Policy	<a href="#">Permissions Policy</a> is a new header that allows a site to control which features and APIs can be used in the browser.

## Warnings

Site is using HTTP	This site was served over HTTP and did not redirect to HTTPS.
--------------------	---

## Upcoming Headers

Cross-Origin-Embedder-Policy	<a href="#">Cross-Origin Embedder Policy</a> allows a site to prevent assets being loaded that do not grant permission to load them via CORS or CORP.
Cross-Origin-Opener-Policy	<a href="#">Cross-Origin Opener Policy</a> allows a site to opt-in to Cross-Origin Isolation in the browser.
Cross-Origin-Resource-Policy	<a href="#">Cross-Origin Resource Policy</a> allows a resource owner to specify who can load the resource.

## Additional Information

Server	This <a href="#">Server</a> header seems to advertise the software being run on the server but you can remove or change this value.
Set-Cookie	This is not a <a href="#">SameSite Cookie</a> .

## Test Scores

Test	Pass	Score	Reason	Info
<a href="#">Content Security Policy</a>	✗	-25	Content Security Policy (CSP) header not implemented	<a href="#">i</a>
<a href="#">Cookies</a>	✓	0	All cookies use the <code>Secure</code> flag and all session cookies use the <code>HttpOnly</code> flag	<a href="#">i</a>
<a href="#">Cross-origin Resource Sharing</a>	✓	0	Content is not visible via cross-origin resource sharing (CORS) files or headers	<a href="#">i</a>
<a href="#">HTTP Public Key Pinning</a>	—	0	HTTP Public Key Pinning (HPKP) header cannot be set, as site contains an invalid certificate chain (optional)	<a href="#">i</a>
<a href="#">HTTP Strict Transport Security</a>	✗	-20	HTTP Strict Transport Security (HSTS) header cannot be set, as site contains an invalid certificate chain	<a href="#">i</a>
<a href="#">Redirection</a>	✗	-20	Does not redirect to an HTTPS site	<a href="#">i</a>
<a href="#">Referrer Policy</a>	—	0	Referrer-Policy header not implemented (optional)	<a href="#">i</a>
<a href="#">Subresource Integrity</a>	—	0	Subresource Integrity (SRI) is not needed since site contains no script tags	<a href="#">i</a>
<a href="#">X-Content-Type-Options</a>	✗	-5	X-Content-Type-Options header not implemented	<a href="#">i</a>
<a href="#">X-Frame-Options</a>	✗	-20	X-Frame-Options (XFO) header not implemented	<a href="#">i</a>
<a href="#">X-XSS-Protection</a>	✗	-10	X-XSS-Protection header not implemented	<a href="#">i</a>

## ENUMERATION

I performed service enumeration to discover information about the services provided by AltoroJ such as port scanning with Nmap to determine which services were open

```
(kali㉿kali)-[~]
$ nmap -p 1-65535 127.14.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-16 10:35 EDT
Nmap scan report for altoro (127.14.0.1)
Host is up (0.000056s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
9090/tcp   open  zeus-admin

Nmap done: 1 IP address (1 host up) scanned in 1.24 seconds

(kali㉿kali)-[~]
$
```

```

$ nmap -Pn --script vuln 127.14.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-16 10:45 EDT
Nmap scan report for altoro (127.14.0.1)
Host is up (0.000063s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
| http-csrf: 32138654 KB in, 2854899 KB out (10140.7 KB/s)
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=altoro
| Found the following possible CSRF vulnerabilities:
| total (1.2 req/conn)
| res Path: http://altoro:80/
| http Form id: frmsearch
| _ Form action: /search.jsp
| _http-internal-ip-disclosure: ERROR: Script execution failed (use -d to debug)
| http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http://ha.ckers.org/slowloris/
| _http-dombased-xss: Couldn't find any DOM based XSS.
| _http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
| /login.jsp: Possible admin folder
| /examples/: Sample scripts
| /login.jsp: Login page
| _/docs/: Potentially interesting folder
9090/tcp open zeus-admin

```

## VULNERABILITY ASSESSMENT

The vulnerability assessment was done manually and with the aid of Burpsuite and a few other automated tools such as Nmap

**Vulnerability:** SQL injection

**Vulnerability Explanation:** using dynamic SQL statements controlled by the user without any validation and sanitation could allow an attacker to modify the statement's meaning or execute arbitrary SQL commands

**Mitigation:** Implement Prepared Statements with Parameterized Queries, Implement User Input Whitelisting. Injection attacks remain the most common attacks leveraged against web applications. One of the most effective mitigation strategies for preventing SQL

Injection attacks is the implementation of Prepared Statements with Parameterized Queries.

**Severity: High CEW** <https://cwe.mitre.org/data/definitions/89.html>

## Usage Authentication Bypass

PERSONAL SMALL BUSINESS

### Online Banking Login

Username:

Password:

PERSONAL SMALL BUSINESS INSIDE ALTORO MUTUAL

### Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details:

**Congratulations!**

You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

Click [Here](#) to apply.

Server Status Check | REST API | © 2022 Altoro Mutual, Inc.

*This web application is open source! Get your copy from GitHub and take advantage of advanced features.*

Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not warrant the accuracy, reliability, or completeness of the information on this website. For more information, please go to <http://www-142.ibm.com/software/products/us/en/subcategory/SW10>.

All rights reserved.

No.	Time	Method	URL	Status	Size
587	http://127.14.0.1	POST	/doLogin	200	1000000000
586	http://127.14.0.1	GET	/login.jsp	200	1000000000
585	http://127.14.0.1	POST	/doLogin	200	1000000000
584	http://127.14.0.1	GET	/login.jsp	200	1000000000
583	https://www.google-analytics.com	POST	/collect?v=1&_v=j96&a=44	200	1000000000
582	https://www.google-analytics.com	GET	/analytics.js	200	1000000000
581	https://portswigger.net	GET	/content/images/svg/icons/	200	1000000000
580	https://portswigger.net	GET	/content/images/svg/icons/	200	1000000000
579	https://portswigger.net	GET	/content/images/logos/port	200	1000000000
576	https://portswigger.net	GET	/content/images/svg/icons/	200	1000000000
575	https://portswigger.net	GET	/content/images/svg/icons/	200	1000000000
574	https://portswigger.net	GET	/content/images/svg/icons/	200	1000000000
573	https://portswigger.net	GET	/bundles/public/staticcms.js	200	1000000000

**Request**

1 POST /doLogin HTTP/1.1

2 Host: 127.14.0.1

3 Content-Length: 50

4 Cache-Control: max-age=0

5 sec-ch-ua: "Not A;Brand";v="99", "Chromium";v="96"

6 sec-ch-ua-mobile: ?0

7 sec-ch-ua-platform: "Linux"

8 Upgrade-Insecure-Requests: 1

9 Origin: http://127.14.0.1

10 Content-Type: application/x-www-form-urlencoded

11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664 Safari/537.36

12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;q=0.9

13 Sec-Fetch-Site: same-origin

14 Sec-Fetch-Mode: navigate

15 Sec-Fetch-User: ?1

16 Sec-Fetch-Dest: document

17 Referer: http://127.14.0.1/login.jsp

18 Accept-Encoding: gzip, deflate

19 Accept-Language: en-US,en;q=0.9

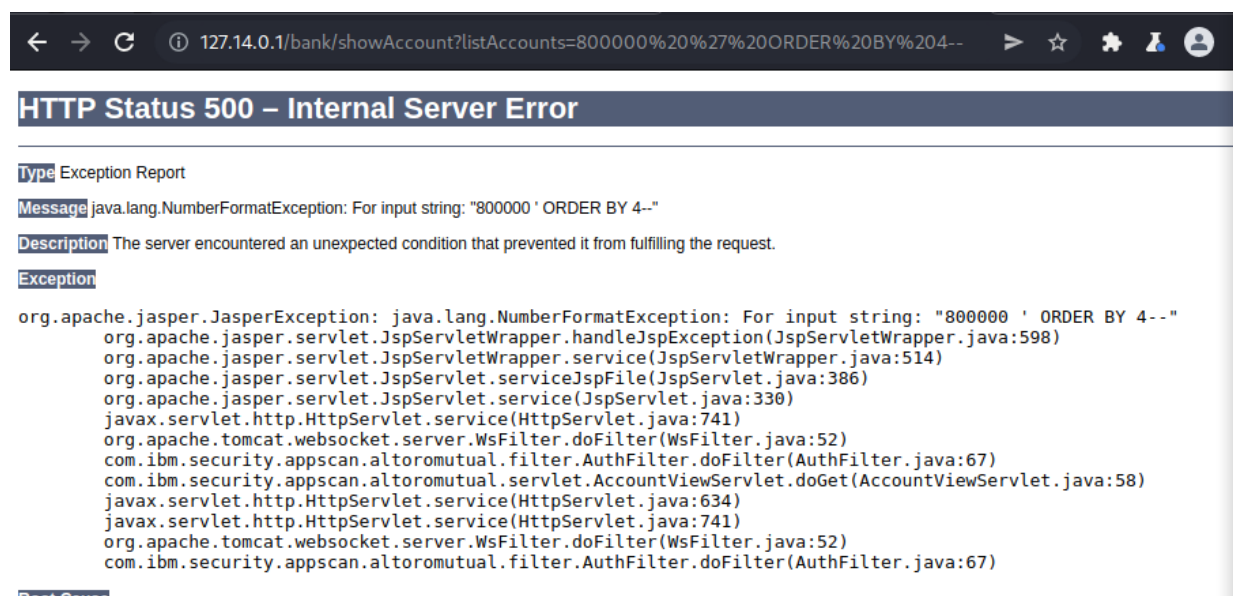
20 Cookie: JSESSIONID=81BC61FF09172DB004ADF93599373D31

21 Connection: close

22

23 uid=admin+%27+OR+1%3D1--&pass=123&btnSubmit=Login

Another vector was in listAccounts but eventually, I did not complete the drill-down



**Vulnerability:** Cross-site scripting (XSS)

**Vulnerability Explanation:** XSS is a security vulnerability (that executes on the client-side) that allows an attacker to compromise the interactions of the web application by allowing the attacker to circumvent the same-origin policy (sop) that is designed to segregate different websites from each other (user data\cookies\information etc..) the malicious (reflected) URL or persistent XSS(stored on the server) will be executed in the context of the victim's web browser doing the action the attacker intended for example carry out any action that the user is able to perform

**Vulnerability Mitigation:** there are a few things you could do to mitigate an XSS vulnerability, first you could filter the user input as strictly as possible (white list) based on what is expected or valid input, it's possible to encode data on output, this will prevent the output data to be interpreted as active content, also it's required to use appropriate response headers using the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend, last you could use Content Security Policy(CSP) to reduce the severity of any XSS vulnerabilities that still occur







Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requested
http://altoro	GET	/		200	9406	HTML	Altoro Mutual		08:14:50 11 M...
http://altoro	GET	/login.jsp		200	8539	HTML	Altoro Mutual		09:04:20 11 M...
http://altoro	GET	/cgi.exe							
http://altoro	GET	/default.jsp							
http://altoro	GET	/default.jsp?content=security.htm	✓						
http://altoro	GET	/feedback.jsp							
http://altoro	GET	/index.jsp							
http://altoro	GET	/index.jsp?content=business.htm	✓						
http://altoro	GET	/index.jsp?content=business_cards.htm	✓						
http://altoro	GET	/index.jsp?content=business_deposit.htm	✓						
http://altoro	GET	/index.jsp?content=business_insurance.htm	✓						
http://altoro	GET	/index.jsp?content=business_lending.htm	✓						
http://altoro	GET	/index.jsp?content=business_other.htm	✓						
http://altoro	GET	/index.jsp?content=business_retirement.htm	✓						
http://altoro	GET	/index.jsp?content=inside.htm	✓						
http://altoro	GET	/index.jsp?content=inside_about.htm	✓						
http://altoro	GET	/index.jsp?content=inside_careers.htm	✓						
http://altoro	GET	/index.jsp?content=inside_contact.htm	✓						
http://altoro	GET	/index.jsp?content=inside_investor.htm	✓						
http://altoro	GET	/index.jsp?content=inside_press.htm	✓						
http://altoro	GET	/index.jsp?content=personal.htm	✓						
http://altoro	GET	/index.jsp?content=personal_cards.htm	✓						
http://altoro	GET	/index.jsp?content=personal_checking.htm	✓						
http://altoro	GET	/index.jsp?content=personal_deposit.htm	✓						

Request	Response
<pre> 1 GET /login.jsp HTTP/1.1 2 Host: altoro 3 Upgrade-Insecure-Requests: 1 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45   Safari/537.36 5 Accept:   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i   mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=   0.9 6 Referer: http://altoro/ 7 Accept-Encoding: gzip, deflate 8 Accept-Language: en-US,en;q=0.9 9 Cookie: JSESSIONID=EC634F4471BD7C35DCP5CF4C0BC7D4FE 10 Connection: close 11 12 </pre>	<pre> 102 &lt;h1&gt;   Online Banking Login &lt;/h1&gt; &lt;!-- To get the latest admin login, please contact SiteOps at 415-555-6159 --&gt; &lt;p&gt;   &lt;span id="ctl0_ctl0_Content_Main_message" style="     color:#FF0066;font-size:12pt;font-weight:bold;"&gt;   &lt;/span&gt; &lt;/p&gt; &lt;form action="/doLogin" method="post" name="login" id="login" onsubmit="return (confirm(input(login));"&gt;   &lt;table&gt;     &lt;tr&gt;       &lt;td&gt;         Username:       &lt;/td&gt;       &lt;td&gt;         &lt;input type="text" id="id1" name="id1" value="" style="width: 150px;" /&gt;       &lt;/td&gt;     &lt;/tr&gt;   &lt;/table&gt;   &lt;input type="submit" value="Login" /&gt; &lt;/form&gt; </pre>

**Vulnerability:** Unencrypted login request

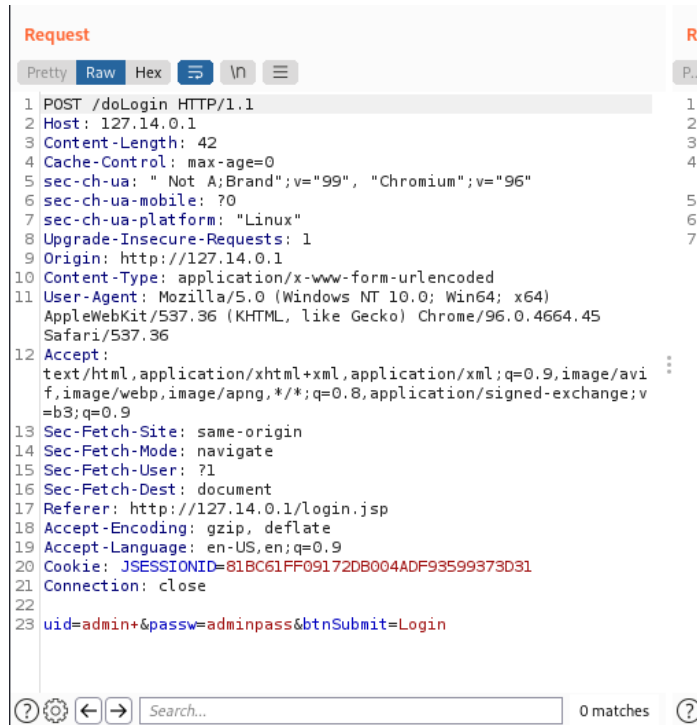
**Vulnerability Explanation:** unencrypted sensitive login information sent over the web could be exposed to a MIM attack

**Vulnerability Mitigation:** encrypt sensitive information before sending it over the web, and use a secure connection such as SSL when sending sensitive information

**Severity:** high

**cwe:** <https://cwe.mitre.org/data/definitions/319.html>

**Usage:**



**Vulnerability:** Error messages with stack traces may expose sensitive information about the application

**Explanation:** information leakage may help an attacker with the reconnaissance stage meaning, An attacker may use the contents of error messages to help launch another, more focused attack for

**Mitigation:** ensure that an error message only the minimal details that are useful to the intended audience and no one else

**Severity - High, Cwe:** <https://cwe.mitre.org/data/definitions/209.html>

**Attack usage:** expose of internal information to leverage current knowledge on the application\server for example an attempt to exploit a path traversal weakness ([CWE-22](#)) might yield the full pathname of the installed application

**Description:** Error messages with stack traces may expose sensitive information about the application.

**Severity:** WARNING

**OWASP:**

**CWE:** CWE-209: Generation of Error Message Containing Sensitive Information

**File:** WebContent/swagger/swagger-ui.js

**Lines:** [1, 1]

Show Code

View File

Not Applicable

False Positive

**File:** WebContent/swagger/swagger-ui.js

**Lines:** [1, 1]

Show Code

View File

Not Applicable

False Positive

**File:** WebContent/swagger/swagger-ui.js

**Lines:** [1, 1]

Show Code

View File

Not Applicable

False Positive

## HTTP Status 500 – Internal Server Error

**Type** Exception Report

**Message** For input string: "0X33"

**Description** The server encountered an unexpected condition that prevented it from fulfilling the request.

**Exception**

```
java.lang.NumberFormatException: For input string: "0X33"
    sun.misc.FloatingDecimal.parseHexString(FloatingDecimal.java:2071)
    sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:1870)
    sun.misc.FloatingDecimal.parseDouble(FloatingDecimal.java:110)
    java.lang.Double.parseDouble(Double.java:538)
    java.lang.Double.valueOf(Double.java:502)
    com.ibm.security.appscan.altoromutual.servlet.TransferServlet.doPost(TransferServlet.java:60)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:660)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
    org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
    com.ibm.security.appscan.altoromutual.filter.AuthFilter.doFilter(AuthFilter.java:67)
```

**Note** The full stack trace of the root cause is available in the server logs.

Apache Tomcat/8.5.43

With this information at hand, I was able to use searchsploit-DB for known CVE related to Apache Tomcat 8.5.43

```
(kali㉿kali)-[~]
$ searchsploit Apache Tomcat 8.5.43
```

Exploit Title	Path
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Executi	jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Executi	windows/webapps/42953.txt

Unfortunately, the application wasn't Vulnerable to the known CVE

```
msf6 > /usr/share/exploitdb/exploits/jsp/webapps/42966.py -u http://127.14.0.1
[*] exec: /usr/share/exploitdb/exploits/jsp/webapps/42966.py -u http://127.14.0.1
```



```
[@intx0x80]

Poc Filename Poc.jsp
Not Vulnerable to CVE-2017-12617
msf6 > █
```

## Additional untested vulnerabilities and attack vectors

**Vulnerability:** unlimited user input size

**Vulnerability Explanation: (untested)** in some cases legit user input size wasn't limited this could lead to exhausting system resources resulting in DoS, and additionally, this could be leveraged to a Heap spraying attack, also if the variable responsible for holding the input size data, for example, is an int its possible to create an int overflow vulnerability and then leverage the attack to different paths

**Vulnerability Mitigation:** limit input size

**Severity:** warning

cwe:<https://cwe.mitre.org/data/definitions/20.html>

<https://cwe.mitre.org/data/definitions/770.html>

With almost similar properties the Regular expression for the user input in some cases combined with the fact that the input is unlimited could result in a denial of service

✓ REGEX DOS - 11

**Description:** Ensure that the regex used to compare with user supplied input is safe from regular expression denial of service.

**Severity:** WARNING

**OWASP:**

**CWE:** cwe-185

**File:** WebContent/swagger/lib/highlight.7.3.pack.js

**Lines:** [1, 1]

Hide Code

View File

Not Applicable

False Positive

```
var hljs=new function(){function l(o){return o.replace(/&/gm,"&amp;").replace(/</gm,"&lt;").replace(/>/gm,"&gt;");}function b(p){for(var o=p.firstChild;o=o.nextSibling)
```

✓ NODE INSECURE RANDOM GENERATOR - 5

**Description:** crypto.pseudoRandomBytes()/Math.random() is a cryptographically weak random number generator.

**Severity:** WARNING

**OWASP:**

**CWE:** CWE-327: Use of a Broken or Risky Cryptographic Algorithm

**File:** WebContent/swagger/lib/marked.js

**Lines:** [740, 740]

Hide Code

View File

Not Applicable

False Positive

```
if (Math.random() > 0.5) {
```

**File:** WebContent/swagger/lib/swagger-oauth.js

**Lines:** [148, 148]

Hide Code

View File

Not Applicable

False Positive

```
var state = Math.random ();
```

---

**Mitigation: change to a stronger random algorithm**

**Attack usage: if an attacker knows the bound of the random function he could try to mimic its results**

Many Headers with security properties were not defined\used in this application such as

- This application does not have API rate limiting controls.**CWE-770: Allocation of Resources Without Limits or Throttling - INFO**
- Description: This application does not have anti CSRF protection which prevents cross-site request forgery attacks. - **CWE-352: Cross-Site Request Forgery (CSRF) - INFO**
- Helmet X Permitted Cross-Domain Policies header is not configured for this application.
- Helmet XSS Protection header is not configured for this application
- Helmet IE No Open header is not configured for this application.
- Helmet Expect CT header is not configured for this application.
- Helmet DNS Prefetch header is not configured for this application
- Helmet Feature Policy header is not configured for this application.
- Helmet X Powered By header is not configured for this application.
- Helmet No Sniff header is not configured for this application.
- Helmet HSTS header is not configured for this application.
- Helmet X Frame Options header is not configured for this application.request
- Helmet Referrer-Policy header is not configured for this application.
- Helmet Content Security Policy header is not configured for this application
- This application does not have API rate limiting controls.
- **CWE-693: Protection Mechanism Failure**

**Severity: info**

**CONCLUSION**



---

The web application contains many possible attack vectors, These issues should be addressed as soon as possible and I would recommend taking the website off the web until the production of a robust web application with additional security features even at the cost of damaging the customer's immediate usage