Assignment 4 Final Report

SOSEC HT2020

Group 01:
Vivek Tejas Rao
Tony Pinberg
Love Nilsson Strand
Alex Salvo
Li Zhang

Department of Computer and Systems Sciences, DSV
Stockholm University

October 20, 2020

Contents

1. Delegation

Member	Responsibilities
Vivek Tejas Rao	To review source code/perform code inspection (Davidson, 2020) with an objective to determine if: 1. passwords are weakly handled 2. weak components within Java tech. are utilized 3. sensitive data such as PII, passwords are susceptible to exposure 4. to explore potential countermeasures for the identified vulnerabilities 5. Running vulnerability analysis on the application (Davidson, 2020) Motivation: Prior experience within Java, Un*x programming
Tony Pinberg	Review jforum structure (Davidson, 2020) and code focusing on insecure configuration issues related to the potential of XSS attacks. Some experience with script- & OO php, java, JS, jQuery, python, MySql, SQLite, AJAX and MVC structure.
Love Nilsson Stand	Reviewing the code (Davidson, 2020) to identify possible flaws exposing the system to SQL injection by reviewing input validation from various input fields like the URL, search field and user profile field and provide assistance to group members that need help. Prior programming experience with C, .NET and Moderate Java experience with some experience of MySQL. Final report: Testing SQL-injection, reviewing the results of static analysis and writing corresponding parts in the report.
Alex Salvo	To review at the code in the Jforum source code (Davidson, 2020) that handles: 1. Password handling. 2. Sensitive data. Have very basic experience of Java from SUPCOM. For the Final report: Used the nmap penetration tool, wrote parts in architectural outline of jforum and tools.
Li Zhang	To review the source code (Davidson, 2020) to identify weaknesses on passwords handling and sensitive data protection. Test static analysis and pentest tools, document and discuss the results obtained from the tools. Basic programming experience with Java (from SUPCOM) and Python (by self-study).

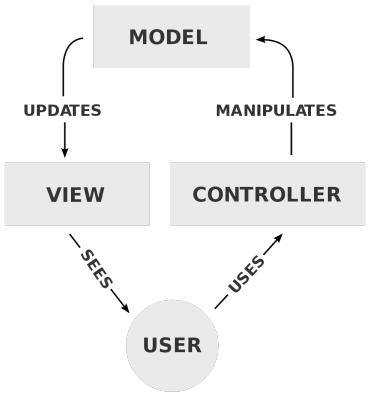
2. Outline of Jforum Architecture

2.1 MVC

Looking at the jforum (Davidson, 2020) package content, at the root level we can see the architecture overview depicted in two different file format versions: jforum-src/forum_model.gif (or .vsd). This structure seems to be a UML diagram, which is the commonly preferred representation for software design (Perjons, 2019). The website documentation also clearly states that jforum is based on an MVC framework. This means we have an architectural pattern and general design established.

The overall purpose of MVC architecture can at its core be described as the separation of input logic, data and presentation (Wikipedia, 2019). Highly modular in nature, MVC is well suited for building complex structures such as a forum, or a web shop (HCL Commerce, 2019).

Allowing users to interact directly with source code – or databases – is generally not a good idea in any scenario. MVC addresses that by introducing the concept of separate controllers where the user may define the desired input. These controllers then in turn perform the actual interaction with the functional code of the models, and the user's interactions may temporarily change the data states based on that input (Stack overflow, 2017). The user's actions via controllers do not change the models, just the way in which the results are displayed – via the view modules. This means that the user's input actions are not in direct control of what resulting output occurs, but must first be appropriately executed and interpreted by the model.



MVC model (source: Wikipedia, 2020)

Request response handling

The code authors (Davidson, 2020) have implemented an interface class for REQUEST where the following arguments are fetched:

- url
- query string
- header
- cookies
- remote address
- port number
- scheme
- server name
- module name
- action name
- user name (depends on browser)

The RESPONSE handling is also established by means of an interface class which returns (among other parameters) (Davidson, 2020) :

- boolean result
- response header
- cookie
- output stream using getOutputStream() which is vulnerable according to the owasp code review (Conklin et al., 2017)

How access control and sessions are achieved

The application (Davidson, 2020) implements the Role Based Access Control (RBAC) security pattern. The end user is assigned a role. Upon issuing a petition to access a resource, the access control engine checks the value of the role and makes a decision regarding the decision. (Rosado, et al., 2006)

The application (Davidson, 2020) uses random.nextint() routine to generate a random number between 0 and 99999999 and hashes it which is then assigned to the session. The session ID is displayed in the URL the first time a user accesses any page within the application (Davidson, 2020).

2.2 FreeMarker HTML

FreeMarker template engine is applying the MVC pattern. It relates to Jforum that this pattern uses a java object that is, for example, setName as a prepared statement inserted into a template format. The apache FreeMarker template engine generates an HTML output with the object together with the HTML (<#FREEMARKER>, 2020) An example in (<FREEMARKER>, 2020) showed how it could work:

```
<html>
 Name: $(name)
</html>
```

The java object that is used in (name) is model.setName ("Karl");

With the FreeMarker engine it generates the output:

```
<html>
Name: Karl
</html>
```

So, in short, this template engine makes it easier for the HTML authors since they do not need to write in the object and can prepare it in the template together with the object and then render it into Jforum.

FreeMarker applies the MVC pattern but is not bound by servlets, a java class that responds to HTTP (<FREEMARKER>, 2020) (Oracle.com, 2010).

2.3 Database Backend

To communicate with the database Jforum uses a pattern called Data access object or DAO. Which serves as an interface between the database and the object represented in the database (Tutorialspoint, 2020).

In Jforum for example they have the User object that doesn't directly communicate with the database, and then the UserDAO that states the functions that the user data access object should have and then GenericUserDAO that implements that functionality and actually communicates with the database. Understanding this shows that this is where it is important to search for weaknesses against SQL-injection because this is

where the queries to the database are written.

3. Description of vulnerabilities / issues found by means of Code Inspection

3.1 Sensitive Data Exposure

Stack Trace Disclosure

Multiple instances of printstacktrace() function/routine was identified in the application (Davidson, 2020) in order to respond to exceptions. See section in appendices for more in-depth details.

Threat agents can cause the application to enter an unknown state which will reveal sensitive information (Stack Trace Disclosure (Java), n.d.)

Potential Mitigation: Use loggers instead (`https://rules.sonarsource.com/java/RSPEC-1148`).

Weak Hashing Algorithm

The given application (Davidson, 2020) hashes the passwords by utilizing MD-5 hash algorithm. MD-5 produces a 128 bit hash code (MD5, n.d.) which is considered weak ("CERT/CC Vulnerability Note VU836068," n.d.). Therefore, sensitive data such as passwords are at a risk of being exposed by means of various attack patterns such as utilizing a rainbow table.

Potential Mitigation: Use strong hashing algorithms such as SHA-512 (Wikipedia SHA-2, 2020) (SolarWinds MSP, 2019).

Storing passwords in plaintext

We suspect that if a user is added through user rest api (see lines 54-96 of UserREST.java), the password for the user would be stored without being encrypted or hashed first (see line 85 of UserREST.java). That is different from the insertSave method in UserAction class, where the password is hashed by MD5 (see line 311 of UserAction.java in package net.jforum.view. forum).

We are not sure about when and how the UserREST.java code will be used and whether it is a real vulnerability. Since neither Covertiy nor Fortify indicate this problem in their static analysis reports, it looks like we may have some misunderstandings on this issue.

3.2 Broken Authentication

Permits brute force

There is no mechanism to limit login attempts in java codes that are responsible for login authentication (LoginAuthenticatior.java, DefaultLoginAuthenticator.java). It is convenient for attackers to perform brute force attacks to steal users or even admin users' accounts.

Potential counter measure: limit failed login attempts or limit logins to a specified IP address or range (`https://phoenixnap.com/kb/prevent-brute-force-attacks`).

Weak password policy

The application (Davidson, 2020) did not enforce any restrictions on the quality of the password in the source code. The application does not refer to commonly used/default password choices to warn the users about their choices. Any user supplied string is accepted as a password and then hashed using a weak algorithm of MD-5

Potential Counter Measure: The application should include condition checks to ensure a strong password. Common restrictions could includes checks on password length, patterns, commonly used passwords or reference with (Haviebeenpwned.com, 2020).

Test observation

During testing of the application (Davidson, 2020) we created a profile for the forum, and noticed that we could use between 1 to 25 characters. The vulnerability is that we were allowed to use; for example, three characters for the password. according to OWASP, this is a serious issue since it does not enforce a firm password policy and such a password could be targeted in a brute force attack (Wichers & Williams, 2017).

Another possible issue we discovered is that we could create a password with an unlimited amount of characters in the password field when we made another profile. The password table truncated it down to 25 characters but did not warn or display an error-message that the password was too long. It could be an issue for attacks. When testing the system we noticed that the session id is posted in the URL in cleartext when first generated. It might allow session hijacking in some situations.(Wichers, D. & Williams, 2017)

3.3 Vulnerable Libraries

The referenced libraries utilized by the application (Davidson, 2020) are outdated. Older versions are being used which provides opportunities for the adversary to exploit these vulnerabilities (Wichers & Williams, 2017). Specification version numbers of the libraries were found by examining the corresponding manifest files. More information can be found in the appendix.

Potential Solutions: Utilize configuration management system to manage patches for libraries (Wichers & Williams, 2017)

3.4 SQL-Injection

After inspecting the parts of the code (Davidson, 2020) that handle input we could conclude that most parts use prepared statements to protect against sql injection. But there was two parts that did not use a prepared statement but instead mixed MySql and code in a way that is vulnerable to SQL injection:

Input field for email for in the "I lost my password" page

```
rs = s.executeQuery ("SELECT username FROM jforum_users WHERE user_email = '" +
email + "'");
```

Found in GenericUserDAO.java line 875.

Updating a post when pressing edit post

```
    s = JForumExecutionContext.getConnection().createStatement();
    s.execute ("UPDATE jforum_posts_text SET post_text = '" +
       post.getText() +
       "', post_subject = '" +
       post.getSubject() +
       "' WHERE post_id = " +
       post.getId() +
       ")");
```

Found in GenericPostDAO.java line 242.

These two weaknesses were also confirmed by testing. We successfully injected sql into the database through both the recover password email field and the update post field.

We also looked at url UrlPattern.java to see how they handle URL input because sql injection in the URL is a common vulnerability.

It appears to divide the input into an array of characters and then uses Trim() to remove leading and trailing spaces between each character, does this prevent sql injection?

After Searching internet for "using trim to prevent sql injection" to see if it does. It appears that it does not as per security.stackexchange.com it appears you can use unicode for spaces instead of actual spaces. For example /**/, %00, %09, %0a, %0d (2020). We suspected that this could leave the program vulnerable to blind sql injection but were unable to verify this by testing it.

During testing we also noticed that the input is not sanitized at all when typing in username, which means that we could create a user with a name like bob'); DROP TABLE users ;– although the sql injection does not work because that part of the code uses prepared statements it is still not a username you want in your database.

Exploiting the vulnerability

Although performing an SQL-injection is easy in theory getting it to do exactly what you want requires knowledge of MySQL syntax that this group lacks, after having spent a few hours testing various injections we only got fairly harmless things to work, like inputting the following in the email field of recover password:

```
‘ or ‘1=1
```

got the system to send a password recovery email to an empty email address, which is quite useless as an attack but it proved that it was possible to input SQL logic as a user which means that with more experience and time we might be able to steal an account or similar.

Similarly we were able to change the heading from the post text when editing a post by inputting the following in the post field:

```
hej’
post_subject = ‘changed from the wrong place’ --
```

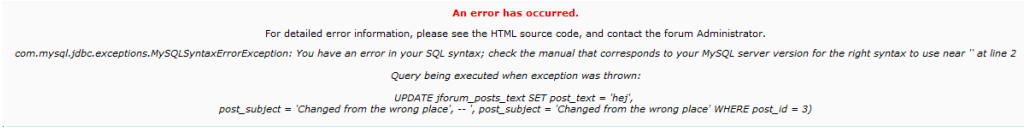
This changes the post subject by injecting sql in the post field instead of by updating the subject field which is completely meaningless but serves as a proof that it is possible to insert sql into the database.

Suggested fixes

For the SQL-injection vulnerabilities the best solution would be to use prepared statements for those parts of the system as well, but it would also be a good idea to sanitize the input by rejecting symbols like “;”, “,” “.” e.t.c for the user name and email. These symbols should provoke an error message saying these symbols are not allowed instead of printing the e.stacktrace error message like it currently does. For the posts you can’t sanitize the input completely so prepared statements is enough.

Reflection

SQL injection corresponds to the OWASP top 10 A1 weakness injection. This weakness is very serious as it allows anyone to inject sql in to the database, because the code uses the executeQuery and execute it appears that you can’t use stacked queries to break out of one query to run any query you want on other parts of the database which limits the amount of damage that can be done, although it could probably be possible to work around this with sufficient knowledge of SQL.



When trying sql injections we also got these very helpful error messages telling us what mistakes we had made, which did make the testing a lot easier.

3.5 XSS

XSS comes in 3 categories: reflected (non-persistent), stored (persistent) and DOM (Wichers & Williams, 2017)(Conklin et al., 2017)(Outpost24, 2018). While still an injection type attack, and largely similar to SQL injection, it is not gaining access to the content of a database itself that is the target of XSS (OWASP, 2011). Rather, it inserts scripts together with the input going into a database in order to manipulate the view other users see in their browser (OWASP, 2011). These users have technically not been the targets of an attack – the site has – but are nonetheless affected by it as the script executes and, in some way, modifies the site view.

When performing manual (strictly ocular) code review in terms of XSS, some of the files specified in the team charter stands out as being more relevant. Primarily UrlPattern.java, BBCode.java, BBCodeHandler.java and FileUploadBase.java, and to a lesser extent RESTauthentication.java and Autokeys.java.

A general finding occurring in several places that stood out early as seemingly strange is that html tags and code is included in comment sections. While html tags might be explained by formatting preview boxes in Eclipse, including functional code seems less appropriate. Even if being commented out prevents the code from rendering, it will still be processed in DOM. If it's not intended to be part of the product output it should be removed to prevent being exploited.

UrlPattern.java

- `<moduleName>.<actionName>.<numberOfParameters> = <var 1>,<var n>`

  Translates into: <moduleName>.<actionName>.<numberOfParameters> = <var1>,<var n>. Since <> are already used frequently, the < and > seem like a deliberate attempt to possibly obfuscate rather than a consistent stylistic choice.

- `myModule.myAction.0 =`

  This function expects variables to always be in correct order but does not seem to validate or enforce that to be the case, only that they are generated that way. Thus, a script might forcibly switch places of var1 and var2. The use of "friendly URL's" then masks this from sight.

  Suggested solution: enforcing policy that disallows code or tags in comments. If it's commented out it should be removed instead.

BBCode.java

First of all, what is BBCode? In short it is simplified commands that allows a user to include some basic html formatting to posted content. It is not an absolute and universally defined language, but can be configured to one's needs. That means a matching process takes place in order to translate the desired "command" into rendered effect – which is a really attractive attack vector for XSS. The structure for BBCode function implements the following functions:

- allows for defining tags,
- turns them into regex, (questionable if regular expression is a good idea of a format)
- replaces with matching code,
- removes quotes and
- "always processes annotations regardless of whether or not the target annotation has been found" (Spring.IO, N.D.)[6].

It does however not seem to sanitize, escape or enforce in any way that which has been defined. User "Chase Seibert" on stackoverflow mentions AntiSamy as one potential solution (Seibert, 2009). Instead of regexing, it parses as html and traverses the DOM and sanitizes according to whitelist. The jforum package does have a function in utilities for "makesafehtml" but that seems to be a proprietary solution maintained. It also seems to be a blacklist concept.

Suggested solution: Use whitelist service rather than blacklist, preferably from an external third party rather than proprietary. Resource commitment is likely to be more prioritized by such a vendor. Another - also preferred - solution is to not use BBCode at all. The aesthetic benefits provided are microscopical, while being a big risk for XSS.

BBCodeHandler.java

- `/bb_config.xml`

  Is this appending secure, or can it be accessed by someone wishing to insert new configs?

- // Shall we remove the infamous quotes?

  Doesn't seem like something the coders would include. Does however sound like something a certain lecturer would put there and it does seem quotations are not sanitized for user input as it should be. Same would apply here.

Suggested solution: Despite the lecturer's explicit view that disallowing quotes, apostrophes etc is a lack-luster solution that should not be encouraged, I would still argue that yes - they should be disallowed / removed. The personal satisfaction of an O'Brien being allowed to spell their name properly does not outweigh the potential downsides it brings. That being said, disallowing quotations should not be the only countermeasure. Properly sanitizing input - and even more importantly - escaping the output, should be the top priority.

FileUploadBase.java

- private long sizeMax = -1;

  This means there is no maximum size limit to uploaded files. Not specifically an XSS issue but most certainly not a very good idea.

Suggested solution: Unless you want to provide unlimited hosting for some reason, enforce a maximum size to any uploaded file.

4. Use of Tools

4.1 Static Analysis Tools

Coverity

We followed the given instructions. In the step of choosing analysis options, besides default settings, we checked the box "Enable checkers that find security vulnerabilities"additionally. We were not sure what the other options meant, and in order to prevent the runtime of the program from becoming too long, we did not choose them.

Finally, 152 errors were found, including 114 Defect occurrences and 38 errors found by SpotBugs Checkers (see Appendix I for the complete list).

Many of the problems we found by the manual code review are contained in the list. For example, the SQL injection vulnerabilities (items 140, 141 and 142 of the list) and weak password hash (item 152 of the list). Another interesting finding is that a CSRF vulnerability (listed in OWASP top 10 v2013, but not in the 2017 version since many frameworks started utilizing CSRF defenses) was detected that we had not found during code review.

Fortify

Unlike Coverity that can deal with a package of files one time, the sourceanalyzer of Fortify checks only one file each time. Considering the limit of time resources, we only chose three files to test. They are GenericPostDAO.java, GenericUserDAO.java, and UserREST.java. See Appendix II for the complete results of the checking.

It seems that Fortify is at least not as good as Coverity in detecting vulnerabilities in the first two files. The output of Fortify reports a low risk of SQL Injection resulted by the use of code "Connection.prepareStatement()" which however seems safe since the prepared statement is used, while neglecting the real SQL injection vulnerabilities that found by both Coverity and manual code review. Moreover, it falsely reports the "Password in Comment" problem. We guess it came to this conclusion only based on discovering the use of the keyword "Password" in the comment. In fact, no password really appears in the comment. For results on UserREST.java, see discussion in Subsection 3.1.3.

4.2 Penetration Testing Tools

Burp suite

This tool was used to perform brute force attacks against the Jforum login page. We followed the instructions on `https://alpinesecurity.com/blog/brute-forcing-login-page-with-burp-suite/`. In order to make the task easier, we assumed that the administration username "admin" had been known and chose the Sniper type of attack and manually added some items including "admin123" as payloads. The attack was successful, as it was indicated by the response status "300" when the right payload "admin123" was delivered.

Nessus

For penetration testing, we tried using Nessus to look for vulnerabilities of the virtual machine that hosts the Jforum server. The chosen policy is "Web Application Tests".

One high, two medium and one low level vulnerabilities were found. The high level vulnerability (134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)) and one medium vulnerability (12085 - Apache Tomcat Default Files) are associated with the Apache Tomcat application used by the server.

The other medium one (85582 - Web Application Potentially Vulnerable to Clickjacking) states that the following webpages do not use a clickjacking mitigation response header and contain a clickable event. According to the vulnerability report, this vulnerability "could potentially expose the site to a clickjacking or UI redress attack, in which an attacker can trick a user into clicking an area of the vulnerable page that is different

than what the user perceives the page to be".

```
The following pages do not use a clickjacking mitigation response header and contain a clickable
 event :

  - http://192.168.121.147:8080/examples/jsp/cal/login.html
  - http://192.168.121.147:8080/examples/jsp/checkbox/check.html
  - http://192.168.121.147:8080/examples/jsp/colors/colors.html
  - http://192.168.121.147:8080/examples/jsp/colors/colrs.jsp
  - http://192.168.121.147:8080/examples/jsp/error/err.jsp
  - http://192.168.121.147:8080/examples/jsp/error/error.html
  - http://192.168.121.147:8080/examples/jsp/jsp2/el/functions.jsp
  - http://192.168.121.147:8080/examples/jsp/jsp2/el/implicit-objects.jsp
  - http://192.168.121.147:8080/examples/jsp/num/numguess.jsp
  - http://192.168.121.147:8080/examples/jsp/plugin/plugin.jsp
  - http://192.168.121.147:8080/examples/jsp/sessions/carts.html
  - http://192.168.121.147:8080/examples/jsp/sessions/carts.jsp
  - http://192.168.121.147:8080/examples/servlets/servlet/CookieExample
  - http://192.168.121.147:8080/examples/servlets/servlet/RequestParamExample
  - http://192.168.121.147:8080/examples/servlets/servlet/SessionExample
```

Note that 192.168.121.147 is the IP address of the virtual machine in the local network and 8080 is the port number for the Jforum service.

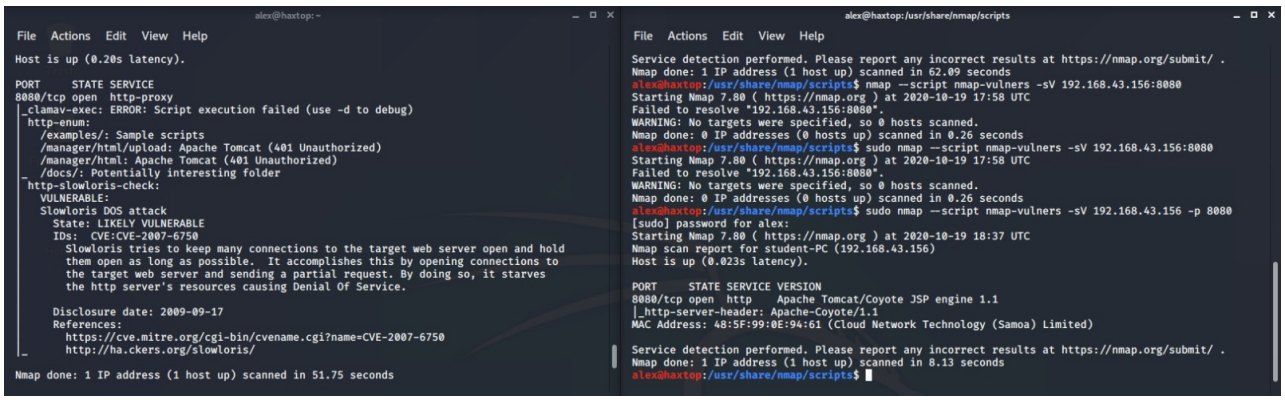
The low level vulnerability (34850 - Web Server Uses Basic Authentication Without HTTPS) is that the server is not using https. This vulnerability is not generated by the flaw in source code of Jforum.

Nmap

Nmap is a free port scanner used to detect servers or/and services within a network (Wikipedia Nmap, 2020). The tool in this assignment was used to detect vulnerabilities and evaluate the jForum security.

The penetration testing evaluation was conducted using two different computers, one with Kali Linux OS installed and one with windows, which has a Virtual box running the jforum platform. Both computers were connected to a router with wireless internet to create a network environment that could enable a nmap scan.

With nmap we did a type of scans: We scan for vulnerabilities against the CVE vulnerability scan script (Borges, 2018) and one scan to determine which services the port was running. In a real attack, this information could exploit vulnerabilities with the version of the software.



Nmap scan resutls

4.3 Reflections

Although our manual audit showed relatively good results as we found some big vulnerabilities. The static code analysis found the same vulnerabilities as us in a fraction of the time in addition to finding other flaws and some bad code practices. Which shows that they are a very powerful tool.

At the same time, the results show that we cannot completely rely on tools (at least cannot just rely on only one tool). For example, the sourceanalyzer of Fortify cannot find the SQL injection vulnerabilities that we found in GenreicPostDAO.java and GenericUserDAO.java through manual code review.

Reference

1. Rao, V. T., Pinberg, T., Strand, L. N., Salvo, A., & Zhang, L. (2020, October 6). SOSEC. Group Assignment 04. Stockholm, Sweden. Retrieved from `https://ilearn2.dsv.su.se/mod/assign/view.php?id=96697`
2. Conklin, L., Robinson, G., Curiel, J., Keary, et. al (2017). OWASP Code Review Guide. Retrieved October 06, 2020, from `https://owasp.org/www-pdf-archive/OWASP_Code_Review_Guide_v2.pdf`
3. Davidson, A. (Ed.). (2020, September 29). Jforum source code. Stockholm, Sweden. Retrieved from `https://ilearn2.dsv.su.se/course/view.php?id=1090`
4. Wichers, D. & Williams, J., 2017. OWASP Top Ten 2017. [Online] Available at: `https://owasp.org` [Accessed 06 October 2020].
5. Stack Trace Disclosure (Java). (n.d.). Retrieved October 13, 2020, from Netsparker: `https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/stack-trace-disclosure-java/`
6. MD5. (n.d.). Retrieved October 13, 2020, from Wikipedia: `https://en.wikipedia.org/wiki/MD5`

7. CERT/CC Vulnerability Note VU836068 [WWW Document], n.d. URL https://www.kb.cert.org (accessed 10.13.20).

8. Outpost24, 2018. "How to identify cross site scripting (XSS) vulnerabilities". https://outpost24.com/blog/How-identify-Cross-Site-Scripting-vulnerabilities

9. OWASP, 2011. "Appsec tutorial series - episode 3: Cross site scripting (XSS). https://www.youtube.com/watch?v=_Z9RQSnf8-g&t=18s

10. Spring.IO, N.D. "Spring Framework". https://docs.spring.io/spring-framework/docs/5.1.7.RELEASE_to_5.1.8.RELEASE/Spring%20Framework%205.1.8.RELEASE/org/springframework/core/annotation/AnnotatedElementUtils.SimpleAnnotationProcessor.html

11. Seibert, C., 2009. "Best regex to catch XSS cross site scripting attack in java". https://stackoverflow.com/questions/24723/best-regex-to-catch-xss-cross-site-scripting-attack-in-java

12. security.stackexchange.com. 2020. Would removing spaces in a string protect against SQL injection?. [ONLINE] Available at: https://security.stackexchange.com/questions/127655/would-removing-spaces-in-a-string-protect-against-sql-injection. [Accessed 13 October 2020].

13. Rosado, D. G., Gutiérrez, C., Medina, . E. . F. & Piattini, M., 2006. A Study of Security Architectural Patterns. u.o., IEEE Computer Society.

14. <FREEMARKER>, 2020. What is Apache FreeMarker?. [Online] Available at: https://freemarker.apache.org/ [Accessed October 20, 2020].

15. Oracle.com, 2010. What is a Servlet?. [Online] Available at: https://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html [Accessed October 20, 2020].

16. Wikipedia, 2020. Nmap. [Online] Available at: https://en.wikipedia.org/wiki/Nmap [Accessed October 20, 2020]

17. Tutorialspoint, 2020. Data Access Object Pattern. [Online] Available at: https://www.tutorialspoint.com/design_pattern/data_access_object_pattern.htm [Accessed October 20, 2020].

18. Borges, E., 2018. SecurityTrails: Top 15 Nmap Commands to Scan Remote Hosts. [Online] Available at: https://securitytrails.com/blog/top-15-nmap-commands-to-scan-remote-hosts [Accessed October 20, 2020].

19. SolarWinds MSP, 2019. SHA-256 Algorithm Overview. [Online] Available at: https://www.solarwindsmsp.com/blog/sha-256-encryption#:~:text=SHA%2D256%20is%20one%20of,sensitive%20information%20using%20SHA%2D256 [Accessed October 20, 2020].

20. Wikipedia, 2020. SHA-2. [Online] Available at: https://en.wikipedia.org/wiki/SHA-2 [Accessed October 20, 2020].

Appendix I: Coverity Errors

Coverity Errors

ID	Checker	File	Line	Function	Classification
1	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	271	net.jforum.SessionFacade.clear()	Unclassified
2	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	242	net.jforum.SessionFacade.getLoggedSessions()	Unclassified
3	BAD_LOCK_OBJECT	src/net/jforum/repository/TopicRepository.java	222	net.jforum.repository.TopicRepository.clearCache(int)	Unclassified
4	BAD_LOCK_OBJECT	src/net/jforum/repository/PostRepository.java	141	net.jforum.repository.PostRepository.remove(int, int)	Unclassified
5	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	328	net.jforum.SessionFacade.isUserInSession(java.lang.String)	Unclassified
6	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	231	net.jforum.SessionFacade.getAllSessions()	Unclassified
7	BAD_LOCK_OBJECT	src/net/jforum/repository/TopicRepository.java	285	net.jforum.repository.TopicRepository.updateTopic(net.jforum.entities.Topic)	Unclassified
8	BAD_LOCK_OBJECT	src/net/jforum/repository/ForumRepository.java	594	net.jforum.repository.ForumRepository.getModeratorList(int)	Unclassified
9	BAD_LOCK_OBJECT	src/net/jforum/repository/ForumRepository.java	687	net.jforum.repository.ForumRepository.getMostUsersEverOnline()	Unclassified
10	BAD_LOCK_OBJECT	src/net/jforum/repository/TopicRepository.java	192	net.jforum.repository.TopicRepository.addAll(int, java.util.List)	Unclassified
11	BAD_LOCK_OBJECT	src/net/jforum/repository/TopicRepository.java	368	net.jforum.repository.TopicRepository.getTopics(int)	Unclassified
12	BAD_LOCK_OBJECT	src/net/jforum/repository/TopicRepository.java	239	net.jforum.repository.TopicRepository.addTopic(net.jforum.entities.Topic)	Unclassified
13	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	123	net.jforum.SessionFacade.add(net.jforum.entities.UserSession, java.lang.String)	Unclassified
14	BAD_LOCK_OBJECT	src/net/jforum/SessionFacade.java	204	net.jforum.SessionFacade.remove(java.lang.String)	Unclassified
15	CALL_SUPER	src/net/jforum/dao/sqlserver/SqlServerPostDAO.java	110	net.jforum.dao.sqlserver.SqlServerPostDAO.selectAllByTopicByLimit(int, int, int)	Unclassified
16	CALL_SUPER	src/net/jforum/dao/sqlserver/SqlServerPostDAO.java	150	net.jforum.dao.sqlserver.SqlServerPostDAO.selectByUserByLimit(int, int, int)	Unclassified
17	CALL_SUPER	src/net/jforum/dao/sqlserver/SqlServerTopicDAO.java	133	net.jforum.dao.sqlserver.SqlServerTopicDAO.selectByUserByLimit(int, int, int)	Unclassified
18	CHECKED_RETURN	src/net/jforum/view/admin/AdminAction.java	207	net.jforum.view.admin.AdminAction.readVersionFromSocket()	Unclassified
19	CHECKED_RETURN	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	287	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.get()	Unclassified
20	CONFIG.MISSING_GLOBAL_EXCEPTION_HANDLER	WEB-INF/web.xml	1	unknown	Unclassified
21	CSRF	src/net/jforum/JForum.java	128	net.jforum.JForum.service(javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)	Unclassified
22	DIVIDE_BY_ZERO	src/net/jforum/view/forum/PostAction.java	1229	net.jforum.view.forum.PostAction.startPage(net.jforum.entities.Topic, int)	Unclassified
23	DIVIDE_BY_ZERO	src/net/jforum/entities/PollOption.java	96	net.jforum.entities.PollOption.getVotePercentage()	Unclassified
24	DIVIDE_BY_ZERO	src/net/jforum/view/admin/UserAction.java	109	net.jforum.view.admin.UserAction.preparePagination(int)	Unclassified
25	DIVIDE_BY_ZERO	src/net/jforum/view/forum/UserAction.java	807	net.jforum.view.forum.UserAction.preparePagination(int)	Unclassified
26	DIVIDE_BY_ZERO	src/net/jforum/util/mail/TopicReplySpammer.java	78	net.jforum.util.mail.TopicReplySpammer.<init>(net.jforum.entities.Topic, net.jforum.entities.Post, java.util.List)	Unclassified
27	DIVIDE_BY_ZERO	src/net/jforum/dao/generic/GenericKarmaDAO.java	144	net.jforum.dao.generic.GenericKarmaDAO.updateUserKarma(int)	Unclassified

Coverity errors report (Part 1)

28	DIVIDE_BY_ZERO	src/net/jforum/view/forum/SearchAction.java	116	net.jforum.view.forum.SearchAction.search(net.jforum.search.SearchOperation)	Unclassified
29	DIVIDE_BY_ZERO	src/net/jforum/dao/generic/GenericKarmaDAO.java	156	net.jforum.dao.generic.GenericKarmaDAO.updateUserKarma(int)	Unclassified
30	DIVIDE_BY_ZERO	src/net/jforum/view/forum/common/TopicsCommon.java	170	net.jforum.view.forum.common.TopicsCommon.prepareTopics(java.util.List)	Unclassified
31	DIVIDE_BY_ZERO	src/net/jforum/view/forum/RecentTopicsAction.java	167	net.jforum.view.forum.RecentTopicsAction.showTopicsByUser()	Unclassified
32	DIVIDE_BY_ZERO	src/net/jforum/view/forum/HottestTopicsAction.java	169	net.jforum.view.forum.HottestTopicsAction.showTopicsByUser()	Unclassified
33	DIVIDE_BY_ZERO	src/net/jforum/view/forum/ModerationAction.java	134	net.jforum.view.forum.ModerationAction.showActivityLog()	Unclassified
34	DIVIDE_BY_ZERO	src/net/jforum/view/forum/ForumAction.java	222	net.jforum.view.forum.ForumAction.show()	Unclassified
35	DIVIDE_BY_ZERO	src/net/jforum/view/forum/PostAction.java	399	net.jforum.view.forum.PostAction.listByUser()	Unclassified
36	DIVIDE_BY_ZERO	src/net/jforum/view/forum/PostAction.java	1301	net.jforum.view.forum.PostAction.delete()	Unclassified
37	DIVIDE_BY_ZERO	src/net/jforum/view/forum/PostAction.java	250	net.jforum.view.forum.PostAction.list()	Unclassified
38	DOM_XSS	templates/default/forum_list.htm	11	<script>	Unclassified
39	FB.DE_MIGHT_IGNORE	src/net/jforum/JForum.java	316	net.jforum.JForum.destroy()	Unclassified
40	FB.DE_MIGHT_IGNORE	src/net/jforum/api/integration/mail/pop/POPConnector.java	108	net.jforum.api.integration.mail.pop.POPConnector.closeConnection(boolean)	Unclassified
41	FB.DE_MIGHT_IGNORE	src/net/jforum/api/integration/mail/pop/POPMessage.java	132	net.jforum.api.integration.mail.pop.POPMessage.extractMessageContents(javax.mail.Message)	Unclassified
42	FB.DE_MIGHT_IGNORE	src/net/jforum/dao/MySQLVersionWorkarounder.java	249	net.jforum.dao.MySQLVersionWorkarounder.loadSqlQueries()	Unclassified
43	FB.DE_MIGHT_IGNORE	src/net/jforum/view/forum/UserAction.java	237	net.jforum.view.forum.UserAction.agreementContents()	Unclassified
44	FB.DLS_DEAD_LOCAL_STORE	src/net/jforum/dao/generic/security/GenericGroupSecurityDAO.java	202	net.jforum.dao.generic.security.GenericGroupSecurityDAO.loadRoles(int[])	Unclassified
45	FB.DLS_DEAD_LOCAL_STORE	src/net/jforum/dao/mysql/security/MySQL323GroupSecurityDAO.java	80	net.jforum.dao.mysql.security.MySQL323GroupSecurityDAO.loadRoles(int[])	Unclassified
46	FB.DLS_DEAD_LOCAL_STORE	src/net/jforum/view/forum/UserAction.java	554	net.jforum.view.forum.UserAction.parseBasicAuthentication()	Unclassified
47	FB.DM_DEFAULT_ENCODING	src/net/jforum/InstallServlet.java	158	net.jforum.InstallServlet.service(javax.servlet.http.HttpServletRequest, javax.servlet.http.HttpServletResponse)	Unclassified
48	FB.DM_DEFAULT_ENCODING	src/net/jforum/JForum.java	285	net.jforum.JForum.handleException(java.io.Writer, net.jforum.context.ResponseContext, java.lang.String, java.lang.Exception, net.jforum.context.RequestContext)	Unclassified
49	FB.DM_DEFAULT_ENCODING	src/net/jforum/api/integration/mail/pop/POPMessage.java	115	net.jforum.api.integration.mail.pop.POPMessage.extractMessageContents(javax.mail.Message)	Unclassified
50	FB.DM_DEFAULT_ENCODING	src/net/jforum/util/MD5.java	75	net.jforum.util.MD5.crypt(java.lang.String)	Unclassified
51	FB.DM_DEFAULT_ENCODING	src/net/jforum/util/legacy/commons/fileupload/FileUploadBase.java	411	net.jforum.util.legacy.commons.fileupload.FileUploadBase.getBoundary(java.lang.String)	Unclassified
52	FB.DM_DEFAULT_ENCODING	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	340	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.getString()	Unclassified
53	FB.DM_DEFAULT_ENCODING	src/net/jforum/view/admin/AdminAction.java	213	net.jforum.view.admin.AdminAction.readVersionFromSocket()	Unclassified
54	FB.DM_DEFAULT_ENCODING	src/net/jforum/view/forum/UserAction.java	222	net.jforum.view.forum.UserAction.agreementContents()	Unclassified
55	FB.DM_DEFAULT_ENCODING	src/net/jforum/view/install/ParseDBStructFile.java	65	net.jforum.view.install.ParseDBStructFile.parse(java.lang.String)	Unclassified

Coverity errors report (Part 2)

56	FB.DM_DEFAULT_ENCODING	src/net/jforum/view/install/ParseDBDumpFile.java	64	net.jforum.view.install.ParseDBDumpFile.parse(java.lang.String)	Unclassified
57	FB.DM_DEFAULT_ENCODING	src/net/jforum/util/legacy/commons/fileupload/MultipartStream.java	454	net.jforum.util.legacy.commons.fileupload.MultipartStream.readHeaders()	Unclassified
58	FB.EQ_DOESNT_OVERRIDE_EQUALS	src/net/jforum/search/SearchPost.java	53	unknown	Unclassified
59	FB.REC_CATCH_EXCEPTION	src/net/jforum/view/install/InstallAction.java	403	net.jforum.view.install.InstallAction.importTablesData(java.sql.Connection)	Unclassified
60	FB.REC_CATCH_EXCEPTION	src/net/jforum/api/integration/mail/pop/POPMessage.java	94	net.jforum.api.integration.mail.pop.POPMessage.extract(javax.mail.Message)	Unclassified
61	FB.RE_BAD_SYNTAX_FOR_REGULAR_EXPRESSION	src/net/jforum/view/install/InstallAction.java	553	net.jforum.view.install.InstallAction.handleDatabasePort(java.util.Properties, java.lang.String)	Unclassified
62	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/view/forum/common/AttachmentCommon.java	378	net.jforum.view.forum.common.AttachmentCommon.makeStoreFilename(net.jforum.entities.AttachmentInfo)	Unclassified
63	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/view/forum/common/UserCommon.java	159	net.jforum.view.forum.common.UserCommon.saveUser(int)	Unclassified
64	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/context/web/WebRequestContext.java	233	net.jforum.context.web.WebRequestContext.handleMultipart(javax.servlet.http.HttpServletRequest, java.lang.String)	Unclassified
65	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	437	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.delete()	Unclassified
66	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	549	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.finalize()	Unclassified
67	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/view/admin/SmiliesAction.java	158	net.jforum.view.admin.SmiliesAction.delete()	Unclassified
68	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/view/forum/common/UserCommon.java	244	net.jforum.view.forum.common.UserCommon.handleAvatar(net.jforum.entities.User)	Unclassified
69	FB.RV_RETURN_VALUE_IGNORED_BAD_PRACTICE	src/net/jforum/view/forum/common/AttachmentCommon.java	321	net.jforum.view.forum.common.AttachmentCommon.editAttachments(int, int)	Unclassified
70	FB.SE_BAD_FIELD	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	138	unknown	Unclassified
71	FB.SE_COMPARATOR_SHOULD_BE_SERIALIZABLE	src/net/jforum/entities/TopicTypeComparator.java	51	unknown	Unclassified
72	FB.SE_NO_SERIALVERSIONID	src/net/jforum/entities/Poll.java	57	unknown	Unclassified
73	FB.SE_NO_SERIALVERSIONID	src/net/jforum/security/PermissionControl.java	55	unknown	Unclassified
74	FB.UCF_USELESS_CONTROL_FLOW	src/net/jforum/cache/JBossCacheListener.java	42	net.jforum.cache.JBossCacheListener.nodeModified(org.jboss.cache.Fqn)	Unclassified
75	FB.UC_USELESS_OBJECT	src/net/jforum/security/XMLPermissionControl.java	234	net.jforum.security.XMLPermissionControl.startElement(java.lang.String, java.lang.String, java.lang.String, org.xml.sax.Attributes)	Unclassified
76	FB.UC_USELESS_VOID_METHOD	src/net/jforum/cache/JBossCacheListener.java	44	net.jforum.cache.JBossCacheListener.nodeModified(org.jboss.cache.Fqn)	Unclassified
77	FORWARD_NULL	src/net/jforum/util/mail/TopicReplySpammer.java	102	net.jforum.util.mail.TopicReplySpammer.<init>(net.jforum.entities.Topic, net.jforum.entities.Post, java.util.List)	Unclassified
78	FORWARD_NULL	src/net/jforum/util/legacy/clickstream/BotChecker.java	59	net.jforum.util.legacy.clickstream.BotChecker.isBot(javax.servlet.http.HttpServletRequest)	Unclassified
79	FORWARD_NULL	src/net/jforum/view/forum/UserAction.java	285	net.jforum.view.forum.UserAction.insertSave()	Unclassified
80	FORWARD_NULL	src/net/jforum/view/forum/PostAction.java	1001	net.jforum.view.forum.PostAction.insertSave()	Unclassified
81	FORWARD_NULL	src/net/jforum/view/forum/PostAction.java	1050	net.jforum.view.forum.PostAction.insertSave()	Unclassified
82	FORWARD_NULL	src/net/jforum/view/forum/PostAction.java	1095	net.jforum.view.forum.PostAction.insertSave()	Unclassified
83	GUARDED_BY_VIOLATION	src/net/jforum/search/LuceneIndexer.java	115	net.jforum.search.LuceneIndexer.createRAMWriter()	Unclassified
84	INSECURE_COMMUNICATION	WEB-INF/config/SystemGlobals.properties	159	unknown	Unclassified

Coverity errors report (Part 3)

85	INSECURE_COMMUNICATION	WEB-INF/config/SystemGlobals.properties	520	unknown	Unclassified
86	LOCK_EVASION	src/net/jforum/JForum.java	243	net.jforum.JForum.checkDatabaseStatus()	Unclassified
87	LOCK_EVASION	src/net/jforum/util/legacy/clickstream/config/ConfigLoader.java	48	net.jforum.util.legacy.clickstream.config.ConfigLoader.getConfig()	Unclassified
88	MISSING_AUTHZ	src/net/jforum/JForumExecutionContext.java	314	net.jforum.JForumExecutionContext.finish()	Unclassified
89	NON_STATIC_GUARDING_STATIC	src/net/jforum/JForum.java	245	net.jforum.JForum.checkDatabaseStatus()	Unclassified
90	NON_STATIC_GUARDING_STATIC	src/net/jforum/JForum.java	246	net.jforum.JForum.checkDatabaseStatus()	Unclassified
91	NULL_RETURNS	src/net/jforum/util/image/ImageUtils.java	89	net.jforum.util.image.ImageUtils.resizeImage(java.lang.String, int, int, int)	Unclassified
92	NULL_RETURNS	src/net/jforum/api/integration/mail/pop/POPMessage.java	66	net.jforum.api.integration.mail.pop.POPMessage.extract(javax.mail.Message)	Unclassified
93	NULL_RETURNS	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	318	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.getString(java.lang.String)	Unclassified
94	NULL_RETURNS	src/net/jforum/view/admin/CategoryAction.java	103	net.jforum.view.admin.CategoryAction.editSave()	Unclassified
95	NULL_RETURNS	src/net/jforum/view/admin/CategoryAction.java	196	net.jforum.view.admin.CategoryAction.processOrdering(boolean)	Unclassified
96	NULL_RETURNS	src/net/jforum/view/admin/ForumAction.java	201	net.jforum.view.admin.ForumAction.processOrdering(boolean)	Unclassified
97	NULL_RETURNS	tests/core/net/jforum/TestCaseUtils.java	98	net.jforum.TestCaseUtils.getRootDir()	Unclassified
98	NULL_RETURNS	tests/core/net/jforum/search/LuceneSearchTestCase.java	226	net.jforum.search.LuceneSearchTestCase.setUp()	Unclassified
99	NULL_RETURNS	src/net/jforum/view/install/InstallAction.java	134	net.jforum.view.install.InstallAction.checkLanguage()	Unclassified
100	NULL_RETURNS	src/net/jforum/dao/generic/AutoKeys.java	126	net.jforum.dao.generic.AutoKeys.executeAutoKeysQuery(java.sql.PreparedStatement, java.sql.Connection)	Unclassified
101	NULL_RETURNS	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	370	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.write(java.io.File)	Unclassified
102	NULL_RETURNS	src/net/jforum/cache/JBossCacheEngine.java	149	net.jforum.cache.JBossCacheEngine.getValues(java.lang.String)	Unclassified
103	NULL_RETURNS	src/net/jforum/search/LuceneContentCollector.java	88	net.jforum.search.LuceneContentCollector.collect(net.jforum.search.SearchArgs, org.apache.lucene.search.Hits, org.apache.lucene.search.Query)	Unclassified
104	NULL_RETURNS	src/net/jforum/util/legacy/commons/fileupload/disk/DiskFileItem.java	338	net.jforum.util.legacy.commons.fileupload.disk.DiskFileItem.getString()	Unclassified
105	NULL_RETURNS	src/net/jforum/view/forum/common/AttachmentCommon.java	315	net.jforum.view.forum.common.AttachmentCommon.editAttachments(int, int)	Unclassified
106	NULL_RETURNS	src/net/jforum/dao/generic/security/SecurityCommon.java	109	net.jforum.dao.generic.security.SecurityCommon.executeAddRole(java.lang.String, int, net.jforum.security.Role, net.jforum.security.RoleValueCollection, boolean, java.lang.String)	Unclassified
107	NULL_RETURNS	src/net/jforum/view/forum/PostAction.java	1403	net.jforum.view.forum.PostAction.downloadAttach()	Unclassified
108	NULL_RETURNS	src/net/jforum/util/image/ImageUtils.java	182	net.jforum.util.image.ImageUtils.saveCompressedImage(java.awt.image.BufferedImage, java.lang.String, int)	Unclassified
109	NULL_RETURNS	src/net/jforum/util/legacy/commons/fileupload/FileUploadBase.java	337	net.jforum.util.legacy.commons.fileupload.FileUploadBase.parseRequest(net.jforum.util.legacy.commons.fileupload.RequestContext)	Unclassified
110	NULL_RETURNS	src/net/jforum/view/forum/common/AttachmentCommon.java	358	net.jforum.view.forum.common.AttachmentCommon.editAttachments(int, int)	Unclassified
111	NULL_RETURNS	src/net/jforum/repository/ForumRepository.java	822	net.jforum.repository.ForumRepository.loadMostUsersEverOnline(net.jforum.dao.ConfigDAO)	Unclassified
112	NULL_RETURNS	src/net/jforum/view/forum/ModerationHelper.java	255	net.jforum.view.forum.ModerationHelper.moveTopics()	Unclassified

Coverity errors report (Part 4)

113	NULL_RETURNS	src/net/jforum/view/admin/PermissionProcessHelper.java	101	net.jforum.view.admin.PermissionProcessHelper.processData()	Unclassified
114	NULL_RETURNS	src/net/jforum/view/admin/ForumAction.java	280	net.jforum.view.admin.ForumAction.insertSave()	Unclassified
115	NULL_RETURNS	src/net/jforum/view/forum/PrivateMessageAction.java	239	net.jforum.view.forum.PrivateMessageAction.sendSave()	Unclassified
116	NULL_RETURNS	src/net/jforum/view/forum/PostAction.java	1443	net.jforum.view.forum.PostAction.downloadAttach()	Unclassified
117	OVERFLOW_BEFORE_WIDEN	src/net/jforum/entities/QuotaLimit.java	70	net.jforum.entities.QuotaLimit.exceedsQuota(long)	Unclassified
118	OVERFLOW_BEFORE_WIDEN	src/net/jforum/entities/QuotaLimit.java	73	net.jforum.entities.QuotaLimit.exceedsQuota(long)	Unclassified
119	RESOURCE_LEAK	src/net/jforum/util/preferences/SystemGlobals.java	187	net.jforum.util.preferences.SystemGlobals.loadDefaults()	Unclassified
120	RESOURCE_LEAK	src/net/jforum/JForum.java	99	net.jforum.JForum.init(javax.servlet.ServletConfig)	Unclassified
121	RESOURCE_LEAK	src/net/jforum/util/preferences/SystemGlobals.java	253	net.jforum.util.preferences.SystemGlobals.saveInstallation()	Unclassified
122	RESOURCE_LEAK	src/net/jforum/dao/oracle/OraclePrivateMessageDAO.java	67	net.jforum.dao.oracle.OraclePrivateMessageDAO.addPmText(net.jforum.entities.PrivateMessage)	Unclassified
123	RESOURCE_LEAK	src/net/jforum/dao/generic/GenericPrivateMessageDAO.java	108	net.jforum.dao.generic.GenericPrivateMessageDAO.addPmText(net.jforum.entities.PrivateMessage)	Unclassified
124	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	256	net.jforum.tools.phpbb2jforum.Main.cleanTables()	Unclassified
125	RESOURCE_LEAK	src/net/jforum/util/preferences/SystemGlobals.java	212	net.jforum.util.preferences.SystemGlobals.loadAdditionalDefaults(java.lang.String)	Unclassified
126	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	237	net.jforum.tools.phpbb2jforum.Main.getTotalPosts()	Unclassified
127	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	287	net.jforum.tools.phpbb2jforum.Main.importTables()	Unclassified
128	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	313	net.jforum.tools.phpbb2jforum.Main.setupPermissions()	Unclassified
129	RESOURCE_LEAK	src/net/jforum/view/install/InstallAction.java	484	net.jforum.view.install.InstallAction.dropOracleOrPostgreSQLTables(java.lang.String, java.sql.Connection)	Unclassified
130	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	313	net.jforum.tools.phpbb2jforum.Main.setupPermissions()	Unclassified
131	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	137	net.jforum.tools.phpbb2jforum.Main.importPrivateMessages()	Unclassified
132	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	137	net.jforum.tools.phpbb2jforum.Main.importPrivateMessages()	Unclassified
133	RESOURCE_LEAK	src/net/jforum/dao/generic/security/SecurityCommon.java	113	net.jforum.dao.generic.security.SecurityCommon.executeAddRole(java.lang.String, int, net.jforum.security.Role, net.jforum.security.RoleValueCollection, boolean, java.lang.String)	Unclassified
134	RESOURCE_LEAK	src/net/jforum/dao/generic/AutoKeys.java	147	net.jforum.dao.generic.AutoKeys.executeAutoKeysQuery(java.sql.PreparedStatement, java.sql.Connection)	Unclassified
135	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	107	net.jforum.tools.phpbb2jforum.Main.importPosts()	Unclassified
136	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	107	net.jforum.tools.phpbb2jforum.Main.importPosts()	Unclassified
137	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	201	net.jforum.tools.phpbb2jforum.Main.importUsers()	Unclassified
138	RESOURCE_LEAK	tools/phpbb2jforum/src/net/jforum/tools/phpbb2jforum/Main.java	201	net.jforum.tools.phpbb2jforum.Main.importUsers()	Unclassified
139	RISKY_CRYPTO	src/net/jforum/util/MD5.java	74	net.jforum.util.MD5.crypt(java.lang.String)	Unclassified
140	SQLI	src/net/jforum/dao/generic/GenericPostDAO.java	244	net.jforum.dao.generic.GenericPostDAO.updatePostsTextTable(net.jforum.entities.Post)	Unclassified

Coverity errors report (Part 5)

141	SQLI	src/net/jforum/dao/generic/GenericUserDAO.java	876	net.jforum.dao.generic.GenericUserDAO.getUsernameByEmail(java.lang.String)	Unclassified
142	SQLI	src/net/jforum/dao/generic/GenericPostDAO.java	246	net.jforum.dao.generic.GenericPostDAO.updatePostsTextTable(net.jforum.entities.Post)	Unclassified
143	SWAPPED_ARGUMENTS	src/net/jforum/dao/hsqldb/HsqldbPostDAO.java	68	net.jforum.dao.hsqldb.HsqldbPostDAO.selectAllByTopicByLimit(int, int, int)	Unclassified
144	SWAPPED_ARGUMENTS	src/net/jforum/dao/hsqldb/HsqldbPostDAO.java	76	net.jforum.dao.hsqldb.HsqldbPostDAO.selectByUserByLimit(int, int, int)	Unclassified
145	SWAPPED_ARGUMENTS	src/net/jforum/dao/hsqldb/HsqldbTopicDAO.java	92	net.jforum.dao.hsqldb.HsqldbTopicDAO.selectByUserByLimit(int, int, int)	Unclassified
146	SWAPPED_ARGUMENTS	src/net/jforum/dao/hsqldb/HsqldbUserDAO.java	61	net.jforum.dao.hsqldb.HsqldbUserDAO.selectAllByGroup(int, int, int)	Unclassified
147	SWAPPED_ARGUMENTS	src/net/jforum/dao/hsqldb/HsqldbPostDAO.java	60	net.jforum.dao.hsqldb.HsqldbPostDAO.selectLatestByForumForRSS(int, int)	Unclassified
148	TRUST_BOUNDARY_VIOLATION	src/net/jforum/context/web/WebSessionContext.java	66	net.jforum.context.web.WebSessionContext.setAttribute(java.lang.String, java.lang.Object)	Unclassified
149	UNENCRYPTED_SENSITIVE_DATA	src/net/jforum/api/integration/mail/pop/POPConnector.java	68	net.jforum.api.integration.mail.pop.POPConnector.openConnection()	Unclassified
150	UNLOGGED_SECURITY_EXCEPTION	src/net/jforum/sso/LDAPAuthenticator.java	136	net.jforum.sso.LDAPAuthenticator.validateLogin(java.lang.String, java.lang.String, java.util.Map)	Unclassified
151	UNSAFE_REFLECTION	src/net/jforum/Command.java	114	net.jforum.Command.process(net.jforum.context.RequestContext, net.jforum.context.ResponseContext, freemarker.template.SimpleHash)	Unclassified
152	WEAK_PASSWORD_HASH	src/net/jforum/util/MD5.java	76	net.jforum.util.MD5.crypt(java.lang.String)	Unclassified

Coverity errors report (Part 6)

Appendix II: Fortify Outputs

```
[/opt/sosec-source/jforum-src/src/net/jforum/dao/generic]
[
[CCEED03A8809B70431C72F0D4987D029 : low : SQL Injection : semantic ]
GenericPostDAO.java(81) : Connection.prepareStatement()
[
[63E89ED241DDCC5E93428976C2BBC311 : low : SQL Injection : semantic ]
GenericPostDAO.java(166) : Connection.prepareStatement()

[5EDBB2075DB339023F133D61D5205A22 : low : SQL Injection : semantic ]
GenericPostDAO.java(168) : Connection.prepareStatement()

[B24F100A50754B2F03731232CBB2CF93 : low : SQL Injection : semantic ]
GenericPostDAO.java(202) : Connection.prepareStatement()

[911A03A0C7D3E5E6DD11D4A906AF5078 : low : SQL Injection : semantic ]
GenericPostDAO.java(264) : Connection.prepareStatement()

[31F12485A25D417EE039029A079DE3D4 : low : SQL Injection : semantic ]
GenericPostDAO.java(309) : Connection.prepareStatement()

[D50F799EDC374AE12A36D22EA450DEB1 : low : SQL Injection : semantic ]
GenericPostDAO.java(371) : Connection.prepareStatement()

[5B3ABD271E533CB5F793036B9C57362A : low : SQL Injection : semantic ]
GenericPostDAO.java(403) : Connection.prepareStatement()

[5C7179A02354DCD76B7BBE987407B2D9 : low : SQL Injection : semantic ]
GenericPostDAO.java(433) : Connection.prepareStatement()

[AC7A9999827998A4F4ABE997A2686E38 : low : SQL Injection : semantic ]
[GenericPostDAO.java(464) : Connection.prepareStatement()
[
[[17614EA013183A9673E4A9E7BC203A1D : low : SQL Injection : semantic ]
[GenericPostDAO.java(493) : Connection.prepareStatement()
[
[[7B539538FE5DCD81F2237F829F8C90D8 : low : SQL Injection : semantic ]
[GenericPostDAO.java(524) : Connection.prepareStatement()
[
[[D5C9A75DE6D5DF4FD38A799FFA67631E : low : SQL Injection : semantic ]
[GenericPostDAO.java(243) : Statement.execute()
[
[[06D21B1851F5112B04229B657B59CBB7 : low : Redundant Null Check : controlflow ]
[    GenericPostDAO.java(120) : Dereferenced : editTime
[    GenericPostDAO.java(120) : Compared with null : editTime
[
[[858553F32B7FEF6E9C202BBD3885C839 : low : Poor Error Handling : Overly Broad Throws : structural
[]
[    GenericPostDAO.java(305)
```

Output for checking GenericPostDAO.java

```
[/opt/sosec-source/jforum-src/src/net/jforum/dao/generic]
[
[834ACBEC147A87CFDB0DCCF0734ADF53 : low : SQL Injection : semantic ]
GenericUserDAO.java(99) : Connection.prepareStatement()

[E9A4A6A510083C97014E9AE612D3D441 : low : SQL Injection : semantic ]
GenericUserDAO.java(134) : Connection.prepareStatement()

[827BE1DBABE354C0BED181C1918B75BE : low : SQL Injection : semantic ]
GenericUserDAO.java(148) : Connection.prepareStatement()

[8015EECDB8FB9EA580F133DACE35B53F : low : SQL Injection : semantic ]
GenericUserDAO.java(177) : Connection.prepareStatement()

[6A4BDD491232DA752F4248C074A25E77 : low : SQL Injection : semantic ]
GenericUserDAO.java(251) : Connection.prepareStatement()

[2165AAE764BCE29DC3DE86E3555AB063 : low : SQL Injection : semantic ]
GenericUserDAO.java(272) : Connection.prepareStatement()

[488E5E854FFE57378578E9DC0503096C : low : SQL Injection : semantic ]
GenericUserDAO.java(389) : Connection.prepareStatement()

[CFF90819840C3BDBAE83269BE996B53B : low : SQL Injection : semantic ]
GenericUserDAO.java(410) : Connection.prepareStatement()

[E6C03DABFACF610CC7E59FB73B8AFDEE : low : SQL Injection : semantic ]
GenericUserDAO.java(431) : Connection.prepareStatement()

[4B30AA69D889E8EB63F5B3B5E7D331DC : low : SQL Injection : semantic ]
GenericUserDAO.java(452) : Connection.prepareStatement()

[EDB77B212959E2B8AA6C362C7AEC9D38 : low : SQL Injection : semantic ]
GenericUserDAO.java(474) : Connection.prepareStatement()

[2037A370E5A48153EBEE6D122570F8B9 : low : SQL Injection : semantic ]
GenericUserDAO.java(506) : Connection.prepareStatement()

[1A2B2F0B7A9D261974C5A826894C3D75 : low : SQL Injection : semantic ]
GenericUserDAO.java(513) : Connection.prepareStatement()

[D61F8D0B9882EC89AF5F9C413EAE3A9C : low : SQL Injection : semantic ]
GenericUserDAO.java(578) : Connection.prepareStatement()
[
[714E3A78288A725815CE5A566CD9F87D : low : SQL Injection : semantic ]
GenericUserDAO.java(606) : Connection.prepareStatement()
[
[B1FBBAA9CEDAEFB3A30D9805ABF17CFD : low : SQL Injection : semantic ]
GenericUserDAO.java(631) : Connection.prepareStatement()
```

Output for checking GenericUserDAO.java (part 1)

```
[
[[04E26C5930A4EC551B29E45BD68EAA0D : low : SQL Injection : semantic ]
[GenericUserDAO.java(650) : Connection.prepareStatement()
[
[[7A9DDCF8CA173C732F2E792775AB166A : low : SQL Injection : semantic ]
[GenericUserDAO.java(685) : Connection.prepareStatement()
[
[[E564BFF78C6902CEC5041DDBC0523F3A : low : SQL Injection : semantic ]
[GenericUserDAO.java(715) : Connection.prepareStatement()
[
[[C92BCBB249ABA86FF24F0B0F45CDD82E : low : SQL Injection : semantic ]
[GenericUserDAO.java(749) : Connection.prepareStatement()
[
[[8819868E0C298294F1D720E8C749BD89 : low : SQL Injection : semantic ]
[GenericUserDAO.java(773) : Connection.prepareStatement()
[
[[3BDD440B387BB5431878767F985153AC : low : SQL Injection : semantic ]
[GenericUserDAO.java(798) : Connection.prepareStatement()
[
[[952792C10FEB9768F29477F7A96EE545 : low : SQL Injection : semantic ]
[GenericUserDAO.java(820) : Connection.prepareStatement()
[
[[EEAF87A33D792D4F0CE459C505E376F7 : low : SQL Injection : semantic ]
[GenericUserDAO.java(851) : Connection.prepareStatement()
[
[[BA7618861D95B45DD46E7D6DD24D70FA : low : SQL Injection : semantic ]
[GenericUserDAO.java(903) : Connection.prepareStatement()
[
[[1CB73D003DA402B3D7E76B20B3E688E1 : low : SQL Injection : semantic ]
[GenericUserDAO.java(935) : Connection.prepareStatement()
[
[[180E5CB830374B8AFDDA3510F8AF4909 : low : SQL Injection : semantic ]
[GenericUserDAO.java(964) : Connection.prepareStatement()
[
[[2EF378BC4A845734AEB2F232636EF06F : low : SQL Injection : semantic ]
[GenericUserDAO.java(984) : Connection.prepareStatement()
[
[[AFDFD81C28F81888E29818B3845F9500 : low : SQL Injection : semantic ]
[GenericUserDAO.java(1006) : Connection.prepareStatement()
[
[[C0653E025EF1CAA243DEFB7E87ECA19C : low : SQL Injection : semantic ]
[GenericUserDAO.java(1063) : Connection.prepareStatement()
[
[[8E6F364EB522FDDF441114B064D7DBEB : low : SQL Injection : semantic ]
[GenericUserDAO.java(1085) : Connection.prepareStatement()
[
[[4D842F17F1C409BD81F7300A61AC0DDC : low : SQL Injection : semantic ]
[GenericUserDAO.java(1117) : Connection.prepareStatement()
[
```

Output for checking GenericUserDAO.java (part 2)

```
[AB96658D1597ACDE58A73B4D2557E42C : low : SQL Injection : semantic ]
GenericUserDAO.java(875) : Statement.executeQuery()

[A1D1476DFEC540B3E3F574CE03A86E13 : low : Dead Code : Expression is Always false : structural ]
    GenericUserDAO.java(505)

[1113AAC008246A3F5447EEDC58FBE7B2 : low : Password Management : Password in Comment : structural ]
    GenericUserDAO.java(790)

[8DF0EA25D61000A8346FDF3382F51738 : low : Password Management : Password in Comment : structural ]
    GenericUserDAO.java(812)

[409F8818CF4880962CACE7598FDD669E : low : Password Management : Password in Comment : structural ]
    GenericUserDAO.java(844)
```

Output for checking GenericUserDAO.java (part 3)

```
[/opt/sosec-source/jforum-src/src/net/jforum/api/rest]

[87BE4860BF7116EF6478B36C8642B605 : low : Poor Error Handling : Overly Broad Cat
ch : structural ]
    UserREST.java(44)

[CD9BF8325D3B1E4B67A1CE2AB43D5068 : low : Password Management : Password in Comm
ent : structural ]
    UserREST.java(50)

[3A84DEB406934F36620C0F23DAD9AB78 : low : Poor Error Handling : Overly Broad Cat
```

Output for checking UserREST.java

Appendix III: Nessus Report (Partial)

192.168.121.147

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	1	2	1	26

Scan Information

Start time: Mon Oct 19 08:32:50 2020
End time: Mon Oct 19 08:36:24 2020

Host Information

IP: 192.168.121.147
OS: Microsoft Windows 7 Professional

Vulnerabilities

134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)

Synopsis

There is a vulnerable AJP connector listening on the remote host.

Description

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

See Also

http://www.nessus.org/u?8ebe6246
http://www.nessus.org/u?4e287adb
http://www.nessus.org/u?cbc3d54e
https://access.redhat.com/security/cve/CVE-2020-1745
https://access.redhat.com/solutions/4851251
http://www.nessus.org/u?dd218234
http://www.nessus.org/u?dd772531
http://www.nessus.org/u?2a01d6bf
http://www.nessus.org/u?3b5af27e
http://www.nessus.org/u?9dab109f

Appendix IV: Code Review Sheet

Senstive Data Exposure			
Safe	Suspicious	Vulnerable	Comments on vulnerabilites
net.jforum.security (Whole package)	SSOUtils.java	MD5.java	md5 is a vul. hash algo. it hashes passwords in SSO without salting
LDAPAuthenticator.java	SecurityCommon.java (Catch Exception string)	DefaultLoginAuthenticator.java	uses md5 object to generate hash of the password which is weak. does not use salt
RetmoteUserSSO.java	UserREST.java (Store password in plaintext, see Line 85)	simpleconnection.java	containts printstack trace(). line number 97.
SSO.java		Tpl.java	Contains printstrack trace . Line number 94.
SSOUtils.java		C3P0PooledConnection.java	Printstrack line157
		Adminaction.java	Printstrack line138, 216
		UserAction.java	confuses encryption with hashing, uses md5 hash codes of clear text passwords to store in db. no salting. no password policy, no length checks, no restrictions on common & easy passwords
		log4j	154 matches, could support information leakage acording to OWASP code review guide.
		LoginAuthenticator.java/DefaultLoginAuthenticator.java	No mechenism to limit login attempts. This makes brute force attack easier to perform.

Sensitive Data Exposure

Using vulnerable components / insecure java libraries	
Vulnerable	Comments
org/apache/log4j/	using older version of 1.2.12 even though Log4j 2.13.3 is the latest release of Log4j. https://logging.apache.org/log4j/2.x/
JavaBeans(TM) Activation Framework Specification	using older version of 1.0 even though 1.1.1 is the latest release. https://www.oracle.com/java/technologies/downloads.html
c3p0-0.9.1	using older version, Have relased c3p0-0.9.5.4 https://github.com/swaldman/c3p0/blob/master/src/dist-static/CHANGELOG
javamail api design specification	using older version of 1.3. latest release of 1.5 is available. https://www.oracle.com/technetwork/java/javamail-1-149769.pdf
java api for servelets	using older version of 2.4. later releases exist. https://mvnrepository.com/artifact/javax.servlet/servlet-api
commons lang	using old version of 2.3. latest release of 3.11 available https://commons.apache.org/proper/commons-lang/
lucene search engine by apache	using old version of 2.2.0 . latest release of 8.6.3 available https://lucene.apache.org/core/8_6_3/index.html
freemarker	using old version of 2.3.9. latest version of 2.3.30 available. https://freemarker.apache.org/docs/api/index.html
postgresql-8.0-313.jdbc3.jar	using old version of 8.0-313. lates version is 13 available: https://www.postgresql.org/support/versioning/
jgroups-all-2.2.9-beta2.jar	using old version of 2.2.9-beta2. lates release were 5.0.0 final: https://sourceforge.net/projects/javagroups/files/
jboss-cache-1.2.4	using old version of 1.2.4. Latest relese were 2.3.0: https://sourceforge.net/projects/jboss/files/JBossCache/
quartz- 1.5.1.jar	using old version of 1.5.1. Latest release is 2.3.0 (2.3.4 snapshot): http://www.quartz-scheduler.org/documentation/2.4.0-SNAPSHOT/index.html
concurrent.jar	using old version of 1.3.1. Latest release is 1.3.4 : https://mvnrepository.com/artifact/concurrent/concurrent/1.3.4
jcaptcha-all-1.0-RC2.0.1.jar	using old version , latest release is 1.0-RC6 : https://mvnrepository.com/artifact/com.octo.captcha/jcaptcha-all

Vulnerable Components