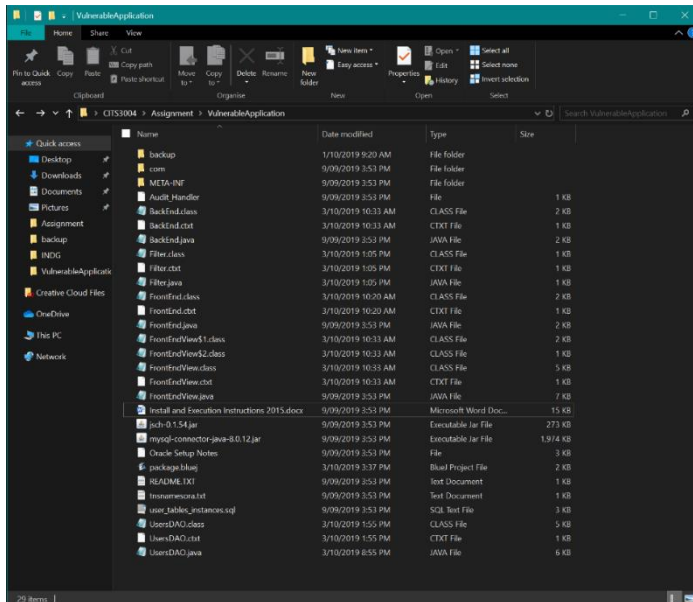


CITS3004 Project: Task 3

Vulnerability 1: Extracting files from the Vulnerable Application

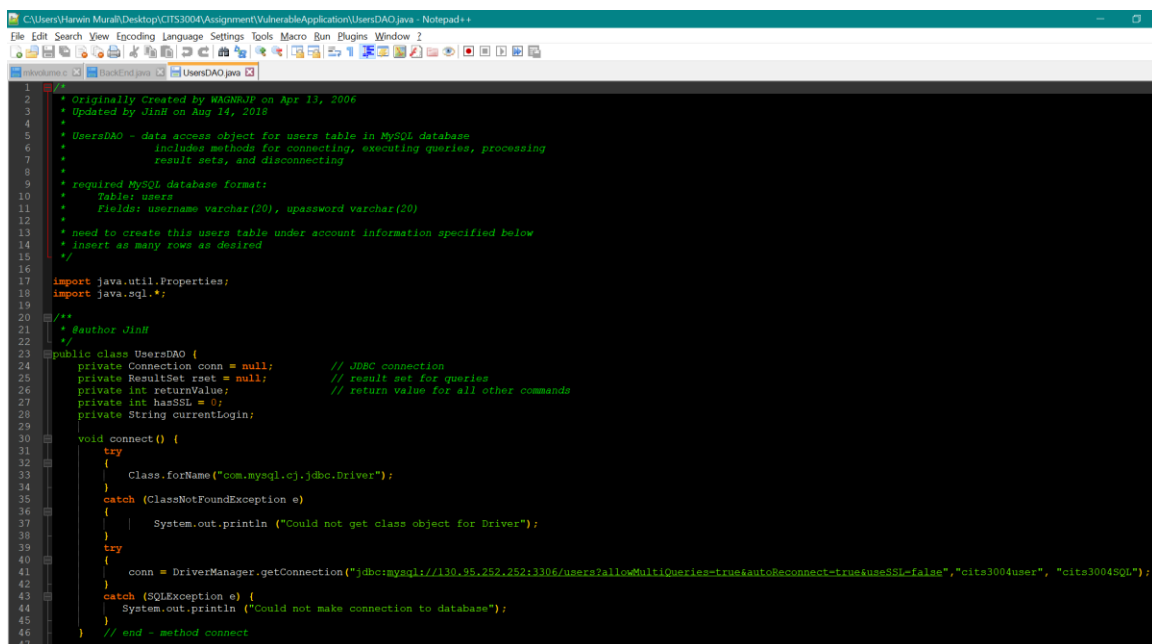
By extracting the VulnerableApplication.jar file, we are able to access the application's contents and most importantly, the source codes of the file. From this, we can identify that BackEnd, Filter, FrontEnd, FrontEndView and UsersDAO are the main components of the file, and also that they were written in Java. A screenshot of the application's contents is provided below:



Vulnerability 2: Analysing the source code

Examining the source codes, we can get an idea as to how the application was constructed and designed to work. From the UsersDAO.java file, we can identify that the VulnerableApplication is running a MySQL database and extract the connection URL for the database

("jdbc:mysql://130.95.252.252:3306/users?allowMultiQueries=true&autoReconnect=true&useSSL=false") and (Username = "cits3004user", Password = "cits3004SQL").



Vulnerability 3: Manipulating source code

```
Statement stmt = null; // SQL statement object

sqlQuery = "select * from users where " +
    "username = " + "'" + fe.getUsername() + "'" + " and " +
    "upasssword = " + "'" + fe.getPassword() + "'";

try {
    stmt = conn.createStatement();
    rset = stmt.executeQuery(sqlQuery);
} catch (SQLException e) {
    System.out.println("Could not execute SQL statement: " + sqlQuery);
}
```

By manipulating the SQL variable template (shown in picture above), hackers can alter the ways queries are processed. For example, hackers could create an additional field that requests for the user's email address. By doing so, they can send phishing emails to the users, potentially tricking them into providing more personal information.

Within the source code, hackers may implement malicious codes that infects the user's system with viruses. Another option would be to implement a keylogger program within the source codes. As an example scenario, a hacker could imitate a lecturer (using the login details extracted from SQL injections, from earlier tasks) and send out an email (spear phishing) to their students with the infected program attached, and requesting them to use this updated version and thus be able to track the keyboard inputs of the users.

Vulnerability 4: Packet Sniffing

No.	Time	Source	Destination	Protocol	Length	Info
160	2019-10-12 16:01:05.2548377	192.168.147.129	130.95.252.252	TCP	74	35546 → 3306 [SYN] Seq=1647020153 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=106219 TSecr=0 WS=128
161	2019-10-12 16:01:05.5458312	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [SYN, ACK] Seq=595860425 Ack=1647020154 Win=64240 Len=0 MSS=1460
162	2019-10-12 16:01:05.5458610	192.168.147.129	130.95.252.252	TCP	54	35546 → 3306 [ACK] Seq=1647020154 Ack=595860426 Win=29200 Len=0
163	2019-10-12 16:01:05.5470804	130.95.252.252	192.168.147.129	MySQL	149	Server Greeting proto=10 version=5.7.23-0ubuntu0.18.04.1
164	2019-10-12 16:01:05.5470206	192.168.147.129	130.95.252.252	TCP	54	35546 → 3306 [ACK] Seq=1647020154 Ack=595860521 Win=29200 Len=0
165	2019-10-12 16:01:05.7445472	192.168.147.129	130.95.252.252	MySQL	289	Login Request user=cits300duser db=users
166	2019-10-12 16:01:05.7451575	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595860521 Ack=1647020389 Win=64240 Len=0
167	2019-10-12 16:01:05.7461498	130.95.252.252	192.168.147.129	MySQL	65	Response OK
168	2019-10-12 16:01:05.7461850	192.168.147.129	130.95.252.252	TCP	54	35546 → 3306 [ACK] Seq=1647020389 Ack=595860532 Win=29200 Len=0
159	2019-10-12 16:01:05.7477150	192.168.147.129	130.95.252.252	MySQL	59	Request Ping, Seq=1647020389, Ack=595860532, Win=29200, Len=0
170	2019-10-12 16:01:05.7478575	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595860532 Ack=1647020394 Win=64240 Len=0
171	2019-10-12 16:01:05.7492312	130.95.252.252	192.168.147.129	MySQL	65	Response OK
172	2019-10-12 16:01:05.7494266	192.168.147.129	130.95.252.252	MySQL	947	Request Query
173	2019-10-12 16:01:05.7495513	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595860543 Ack=1647021287 Win=64240 Len=0
174	2019-10-12 16:01:05.7514227	130.95.252.252	192.168.147.129	MySQL	1138	Response
175	2019-10-12 16:01:05.7688461	192.168.147.129	130.95.252.252	MySQL	72	Request Query
176	2019-10-12 16:01:05.7689570	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595861027 Ack=1647021305 Win=64240 Len=0
177	2019-10-12 16:01:05.7700583	130.95.252.252	192.168.147.129	MySQL	350	Response
178	2019-10-12 16:01:05.7819973	192.168.147.129	130.95.252.252	MySQL	75	Request Query
179	2019-10-12 16:01:05.7821886	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595861923 Ack=1647021326 Win=64240 Len=0
180	2019-10-12 16:01:05.7834122	130.95.252.252	192.168.147.129	MySQL	65	Response OK
181	2019-10-12 16:01:05.7835482	192.168.147.129	130.95.252.252	MySQL	91	Request Query
182	2019-10-12 16:01:05.7836684	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595861934 Ack=1647021363 Win=64240 Len=0
183	2019-10-12 16:01:05.7849783	130.95.252.252	192.168.147.129	MySQL	65	Response OK
184	2019-10-12 16:01:05.7864374	192.168.147.129	130.95.252.252	MySQL	75	Request Query
185	2019-10-12 16:01:05.7864870	130.95.252.252	192.168.147.129	TCP	60	3306 → 35546 [ACK] Seq=595861945 Ack=1647021384 Win=64240 Len=0

Using the Wireshark Packet Sniffer, we are able to capture the network traffic as they are transmitted over a TCP protocol and identify the platform that the database is running on, in this case, MySQL. We can also identify the source and destination ports (35546 and 3306 respectively), and the destination IPv4 address of the MySQL database (130.95.252.252 - consistent with the MySQL database connection URL mentioned above).

Vulnerability 5: Network/TCP Session Hijacking

With the information extracted from packet sniffing earlier, hackers can perform IP Spoofing to hijack the TCP session. Using the IP addresses, hackers can impersonate a different user's identity and send spoofed packets, compromising the integrity of the connection.