CSS 337 Secure Systems
Professor : David LeBlanc
Team Members : Joe Ma, Leo Liu and Mariana Chagoyan.
Assignment: Threat Modeling Process of the Java Chat application

Threat Modeling Process

- Collect Background Information
- Model the System
- Determine Threats
- Determine Vulnerabilities

Collect Background Information
1a. User Scenarios

Normal scenarios:

1. Client1 and Client2 login to the Chat Server with the correct username and password. (Step 1 - Step 4 in DFD Lv1)
2. Client 1 sends message the server, then the server broadcasting it
3. Client 2 sends message the server, then the server broadcasting it

Unusual scenarios:

1. Any client can put the wrong username and password
2. Multiple clients can login to the server with the same username and password

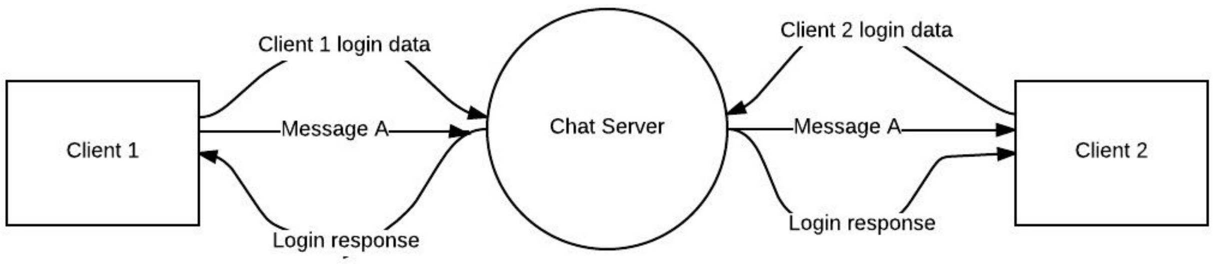
1b. External Dependencies

1. The database will be the file system and it will run on the server . This will include the Java Chat application
2. Application only works over LAN
3. Everyone may have the access to the server program in addition to the administrators

This is a threat model of a Java Chat application, which includes DFD trust boundaries, list of vulnerabilities and mitigations.

2. Model the System
2a. Context Diagram
A DFD of the Java Chat application showing high level data flows between target system and external clients.

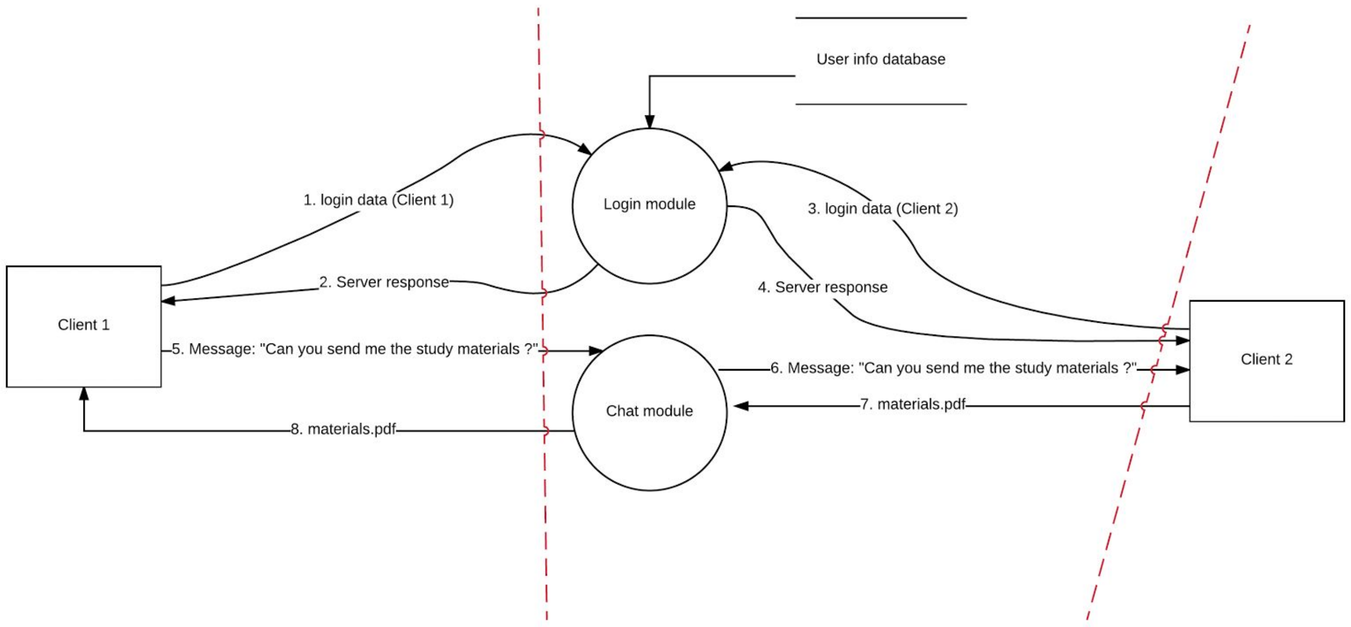
Figure 1



2b. Level 0- DFD

A level 0 DFD of the Java Chat Application showing the major subsystems and the data flows between them and from outside their boundaries.

Figure 2



In the context diagram, there are three parts including 2 clients as entities and 1 server as a process component. The processor handles all data from entities and sends them back correct response.

In the DFD Level 0 diagram, the chat server is separated as two different modules, login modules and chat modules. The login module handles all request related to login issues and the chat modules broadcasts messages from clients with maximum of 50.

The boundaries are set between processors and entities because programs of the server and client are installed in different machines. Both of them are connected through the network.

3. Entry Points

3a. Login Page

A user must first connect to the chat server before they are allowed to do anything else with the messenger.

3b. Login function

A user is able to log into the chat client by typing in their username and password into the provided fields in jMessenger.

3c. UDP

The chat client works through the usage of UDP packets. These packets are sent from a chat client to the server. The server then broadcasts the message to all users in the chat room.

3d. Database

The database is an XML file called "Data.xml". It is located on the machine where the server is first initialized. It stores the usernames and passwords of all users registered to the chat client.

4. Assets

4a. User Login Details

The login credential that a client will use to log into the Java Chat App. This is comprised of a username and password.

4b. User

Assets relating to the users on the chat system: An attacker could potentially impersonate the user. In doing so, other users may reveal personal or secret information to the attacker.

4c. Message

Assets relating to message: Hacker can steal the message content. Packets may be intercepted.

4d. Server Uptime

An attacker can flood server with UDP packets to bring down the system. This is possible because there are no defense systems in place which protects the server from rapid repeated connection attempts.

4e. Database (File System):

The database contains all user information including username and password.

5. Trust Levels

5a. User with Valid Login Credentials

This is a user who has connected to the Java Chat application and is able to log in with valid credentials. Credentials are comprised of a username and password. Usernames and passwords are authenticated by user input and are compared with ones stored in the database. These credentials are used to cross reference with a unique ID number, which has been provided since user registration. In addition, a standard user has the ability to send files and broadcast messages to other users on the chat server. They can also send and receive files.

5b. User with Invalid Login Credentials
This is a user of the Java Chat application who lacks access to the chat server and may be attempting to log in using invalid credentials. They may not even know what port the chat is living on. However, this information could be brute forced.

5c. Database Server Administrator
In the case of this chat application, the database server administrator is the person with access to the computer hosting the server. They have the ability to look through the "data.xml" file to see all user registration data such as username and password.

6 Determine and Analyze Threats

6.1 Threat Categorization

a) Repudiation
This consists of a threat action performing an illegal operation in the system that lacks the ability to trace the prohibited operations. The Java App server can get compromised or attacked by a Denial of service (DoS) attack, and to determine how this actually happened, we need to collect logs of all activity and ensure that logging is working properly.

b) Information Disclosure
This consists of threat action performing an attack by first exploiting the target system to find weakness and to collect as much information. The Java App web server can get targeted if it's exposing just basic information that relates to what type and what version is its web server.

A common source of web server information leak is the server header exposure, which involves each time a user request a resource from a web server, such an image, or sound clip, headers and its values are returned with the requested resource. This information can help the attacker to gather more information in our system and collect more tools to improve his/her attack in our system. Therefore, this can be prevented by limiting the amount of information that could be available to a possible attacker. In particular we can prevent the server header from being exposed in the responses generated by an IIS server and using URL scanner that can remove the server header option. In addition usernames and passwords should not be exposed or easily accessible in the system.

c) Denial of Service
This consists of threat action performing an attack by disrupting the server performance, the applications and services that run in the server, and that will block users from accessing the site.

The Java Chat application can become a target because it has not limits on the amount of packets a client can send to the user. Therefore, when testing if the system is vulnerable, pings are sent to the directed broadcast address of the external networks to see if there is a router misconfiguration. Here we check for outdated chargen and echo services can be conducted because we don't want these services to be abused and be running on external network. Chargen is usually found on port 19, both UDP and TCP and echo is found on port 7 also both UDP and TCP

7. Determine and Analyze Vulnerabilities

All possible attacks should be evaluate from the attacker's point of view and we must consider the entry and exit points.

a) Weak passwords can affect security
If user apply weak passwords, it is very likely that his/her account can get hacked.

b) Passwords are not encrypted
When a database file is stolen, the attacker can simply open up the file to read the username and password.

c) Brute Force Attack
The Chat Server doesn’t limit times that users can login with the invalid username and password. The attacker can attempt to login the system by Brute Force attack with the big dictionary with a large set of username and password.

d) Inadequate logging capabilities
The server does not keep logs of any activity done with the server. While it does display who is connected and from which thread on the jServer, it does not save the information. In addition, chat history has not been implemented yet.

e) Message not encrypted
Messages sent by user are not encrypted so hacker can steal UDP packets by network protocol analyzers like Wireshark.

f) No UDP packet flooding protection
An attacker can flood the server with connection attempts to easily bring down the server.

g) No access level defined in the database
Everyone has the same access to the database so the hacker can manipulate the database manually or by malicious code.

h) Database not encrypted
The contents in the database can be opened by notepad or other editors easily. Hackers can open see what it stores easily

i) File Transfer function reveals the recipient’s IP address to the file sender
Files are transferred by sending ‘upload_req’ as a message. File transfers are accepted by typing in ‘upload_res’. After this, the user who accepted the file request reveal’s the computer’s IP address and the corresponding port which was opened to accept the file transfer.

j) Chat app with virus scanning
Attacker can send malicious software to other users through the application.

Mitigation

a) To increase password complexity, the program should force the user to input special characters into the password field to increase security.

b) To fix the issue with the passwords being stored as plain text, the program should have an encryption module to ensure passwords are secured.

c) To fix the issue of a brute force attack, there should be a setting to block login attempts if there are excessive failed attempts.

d) Since the system logging capability is absent completely, it should be implemented.

e) Messages being sent to and from the server should be encrypted, then decrypted. This is important if the server is used for sensitive information.

f) UDP packet flooding can be combatted with a firewall rule which disallows rapidly repeating connection attempts from the same source.

g) Access levels can be defined. Elevated privileged may be required to edit the contents of the database.

h) The XML file containing the user information can be encrypted to prevent attackers from reading the plain text stored within.

i) Since file transfer works by connecting one client directly to another, the IP address and port of where the file transfer will take place are revealed. To mitigate this risk, the chat application should send the file through the server before it is sent to the final destination. With a middle-man, both user's IP address will remain hidden from each other.

j) Since the attacker can send malicious software through the application, the application should disallow executable files to be uploaded and shared between users. In addition, there could be a service to scan the files for viruses before they are uploaded.

References

1. Threat Modeling.pptx (provided in class)
2. Assessing Network Security, Lam, LeBlanc and Smith, Microsoft Press
3. https://www.owasp.org/index.php/Application_Threat_Modeling#Decompose_the_Application
4. http://www.coc.qu.edu.sa/en/dr.husam.alhamad/documenthusam/gp/samples/332chattingfinalreport.pdf
5. https://www.owasp.org/index.php/Application_Threat_Modeling#Decompose_the_Application