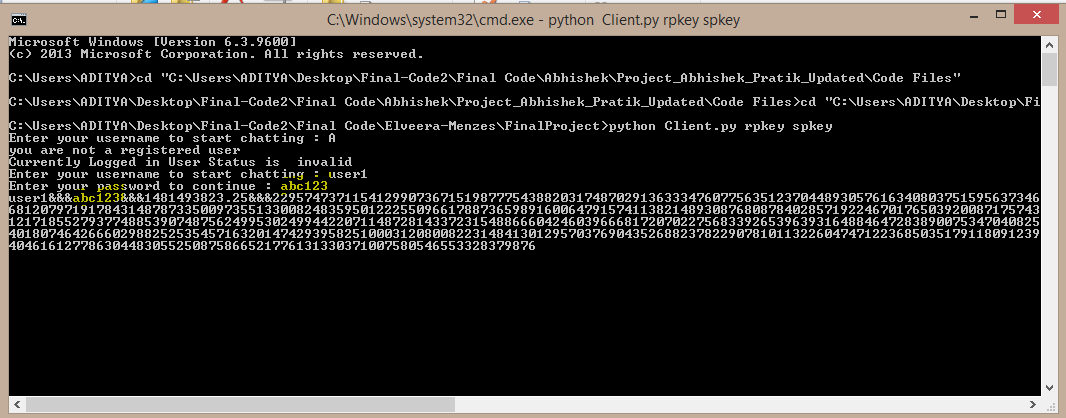
Flaws observed in Designs

|  |  |
| --- | --- |
| Team | Rishabh Jain – Anurag Dwivedy |
| Flaw 1 | **Possible Denial of Service attack** |
| Explanation | Here is the flow   1. When a new client comes online the server gives a puzzle to the client. The server stores the answer in a cookie-map 2. The client then prompts the user to enter the username and password in the terminal. Once the user enters this information, the client sends back the solved puzzle along with username / password encrypted with server public key 3. The server then decrypts and verifies the solved puzzle by looking up the cookie map and then verifies the password and authenticates the client.   The problem here is that after step 3 the cookie map is not cleared and if an intruder captures the client’s response at step 2 and keeps replaying that packet, the server will decrypt the message first and then verify the cookie then sends back an error message saying this user has already logged in. |
| Flaw 2 | Possible Denial of Service attack because of not cleaning the user input |
| Explanation | The messages which are being sent between the clients are not cleansed from user input.  Each message consists of various components such as encrypted message, signature etc, and these components are separated by \*\*\* characters and if a user enters (send Client1 \*\*\*) then the message processing at the Client1’s end fails and an error called Verification failed throws up on the screens (They have handled the error) but a malicious client can play this message in the loop and overload the other client as the client is doing some work in verifying the message. |

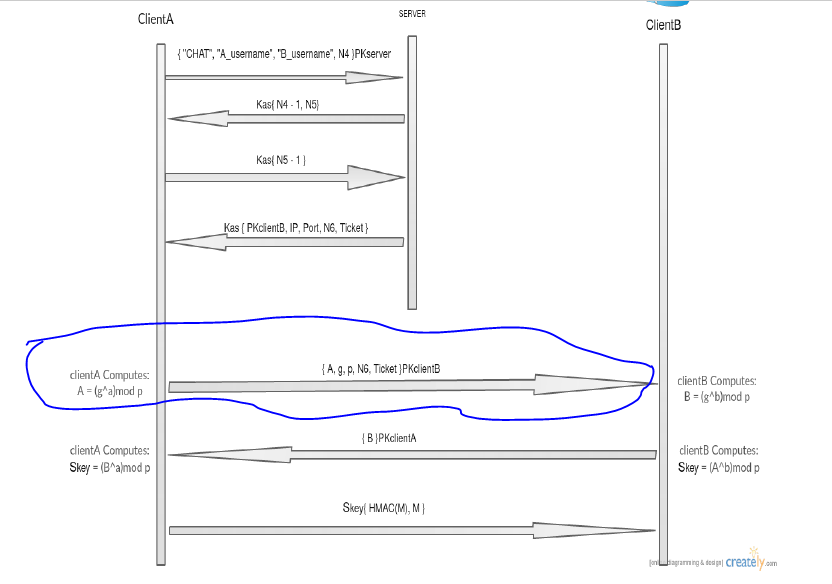
|  |  |
| --- | --- |
| **Team** | Nethra Balasubramanian – Elveera Menezes |
| Flaw | **Password is not encrypted and sent in the open** |
| Explanation | Although the design says, a hash of the password is sent, in the implementation the password is sent in the clear, which is an easy target for eavesdropping and impersonation. |



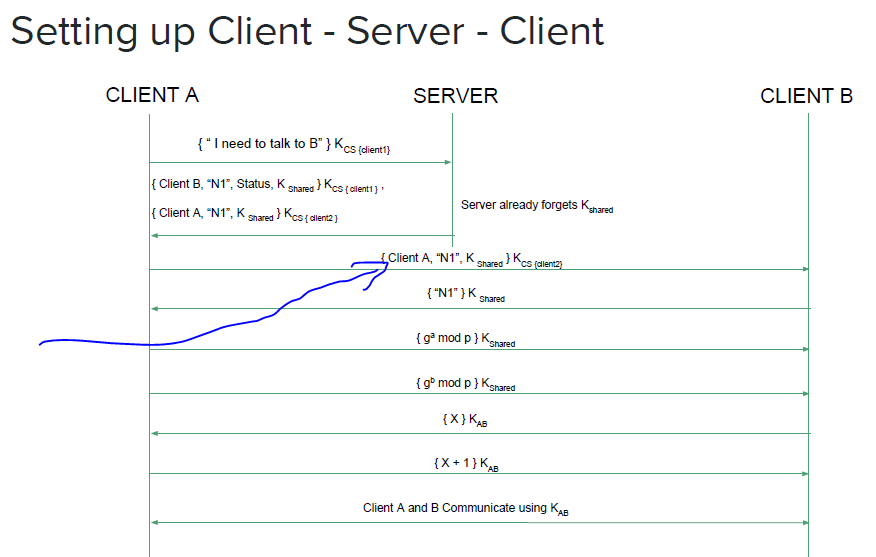
|  |  |
| --- | --- |
| Team | The Mitnick Protocol |
| Flaw | **No Dos Protection** |
| Explanation | The first message which is sent from the Client to the server involves asymmentric encryption of username ,Nonce,Symmetric key and password. An intruder can easily bombard the server with useless packets and make the server perform asymmetric decryption |

|  |  |
| --- | --- |
| Team | Ankeeta Nandal – Akshay Monga |
| Flaw | **Possible DOS attack** |
| Explanation | The first message which is sent to the server by the client is encrypted servers public key . Now the server has to decrypt the message to know what it contains.An intruder can capture the first message sent by a client and flood the server with the first hello message. Or set up threads to just fire the first message at the server. |

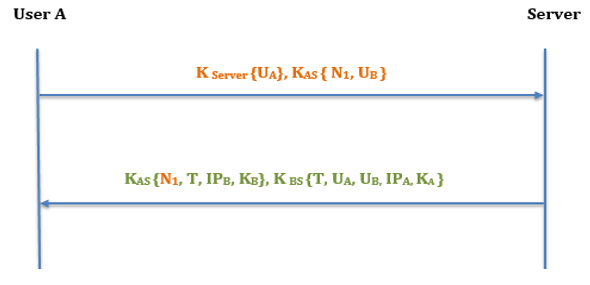
|  |  |
| --- | --- |
| Team | Ankeeta Nandal – Akshay Monga |
| Flaw | **Possible impersonation of Client A to Client B** |
| Explanation | This flaw is due to the fact that when Client A talks to Client B , the Client A talks to the server first ,requesting a common key and a ticket to B. Since the Client B does not provide a nonce ,the freshness of the ticket cannot be confirmed. And in the event the session key between the Server and B is compromised and an intruder has access to this , then he will create his own ticket , which has his public key,ip address and port and username as A. |

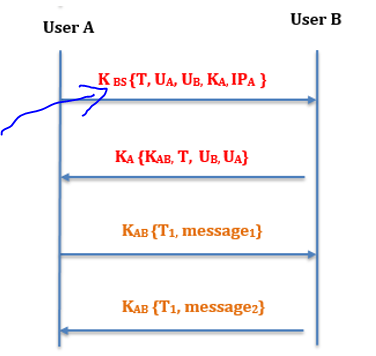


|  |  |
| --- | --- |
| Team | Anup Vikas |
| Flaw | **Possible impersonation of Client A to Client B** |
| Explanation | When Client A wants to talk to Client B , Client A directly approaches the server ,instead of asking Client B a nonce. If the session key between the server and client B is somehow compromised , then the intruder can impersonate A to B. One other thing to be noted both in the above flaw and this is that, the ip address of A is contained in the ticket and B does not verify if the address of the sender and the address in the ticket are same. Because of this any intruder can insert his own ticket and successfully impersonate A to B |

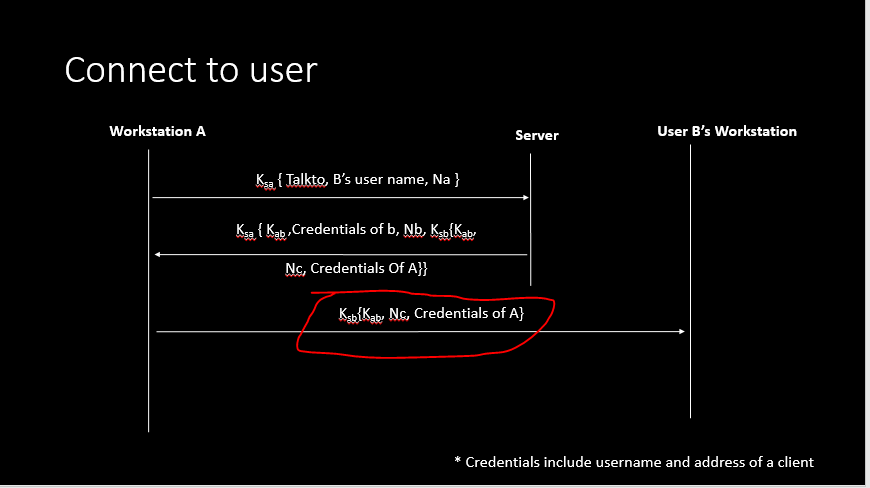


|  |  |
| --- | --- |
| Team | Abhishek Sawarkar – Pratik Pandey |
| Flaw | **Possible impersonation of client A to Client B** |
| Explanation | Again the same flaw as discussed above , in peer to peer communication the server is contacted first ,for the ticket and the freshness of this ticket is not verified. Again if somehow the session key between the server and Client B is compromised, then an intruder can impersonate the Client A to to Client B ,bypassing the server. And again the additional flaw is that the ip address and port of Client A is retrieved from the ticket. The intruder can insert his own ip and port and can start a chat with B as if he is A from his own location. |





|  |  |
| --- | --- |
| Team | Vineeth Venugopal Ravindra – Pushpinder Singh |
| Flaw | Possible Impersonation of Client A to Client B |
| Explanation | Same explanation as the ones above. Client B is not contacted first and hence freshness of the ticket obtained from the serer cannot be confirmed .And because of this if the Ksb is somehow compromised then the intruder can impersonate any user to client B |



|  |  |
| --- | --- |
| Team | Samkeet - Bhavik |
| Flaw | **Possible DOS** |
| Explanation | The first message sent to the server by the client is encrypted with the server’s public key and the server has to decrypt this message. And hence is a possible candidate for a DOS attack. |