# HW2 Location

Christopher Voltz HW2 Location: <https://github.com/chrisvoltz19/AFIT-CSCE689-HW2-S>

# HW2 Note

I have not finished the code yet and will be coming in to discuss where I am stuck.

# HW2 Questions

1. Assess your server/client architecture and identify current risks to Confidentiality, Integrity and Availability
   1. Confidentiality
      1. Don’t have protection on password file
      2. Pass the passwords in the clear (could use Kerberos for more security)
      3. Information leakage (enter a valid username and an invalid password still reveals that the attacker has a valid username)
      4. Doesn’t use encryption (hashing the passwords helps keep them secure particularly with a good salt)
   2. Integrity
      1. IP’s can be spoofed and there isn’t any protection in this system against that
      2. The writing methods don’t have much security to protect integrity checks
      3. The files could be edited by hand ruining their integrity
   3. Availability
      1. Whitelisting can severely limit availability for new users if the admin takes too long or something happens to the list.
   4. All three parts of CIA could be compromised fairly easily so the risks are currently high.
2. Identify at least three security techniques from your reading that you would implement on your server to make it more secure.
   1. Utilize Kerberos to better protect passwords in transit (provide secure channels)
   2. Add more secure servers to counter potential Denial of Service (DOS) attacks
   3. Add different levels of rights (admin, user, etc.) that are allowed to perform differently functionality
   4. Add a firewall to inspect traffic
   5. Utilize two-factor authentication
   6. Moving toward a zero-trust network
3. Besides implementing the algorithm, what resources would you need to implement Kerberos authentication in your architecture?
   1. To implement Kerberos, I would need a system to act as the authentication server and a system to act as the ticket granting service, in addition to the algorithm.
   2. The authentication server authenticates a user and provides a key that is used for setting up secure channels with servers.
   3. The ticket granting service sets up secure channels by issuing tickets that allow the server to check if the client is who they say they are.
4. If you were going to implement authorization management to fine-tune user access to information, how would you do it?
   1. The book discusses how a traditional method of authorization management where the users are given permissions/rights for what they can do on the system. However, in a distributed system, this is difficult to extend because it would require accounts for each user on each machine. This would rapidly grow as the system expanded. It instead proposes using capabilities. A capability is an unforgeable data structure for a resource that specifies exactly the access rights that the holder of the capability has with respect to that resource. Since it is better able to be applied to distributed systems, it seems the better choice in this instance. Additionally, tickets are easier to pass and delegate rights.
   2. Attribute certifications are a potential implementation. They are used to list (attribute, value) pairs that apply to an entity. A specific use is to list the access rights the holder of the certificate has with respect to the identified resource.
   3. It is important to note that the drawback with this strategy is that certificates are difficult to revoke once they have been issued.