In *A Hybrid Technique For SQL Injection Attacks Detection and Prevention* and *A Classification of SQL Injection Attacks and Countermeasures* both of these papers address the severity SQL Injection attacks, although they focus on different aspects, these papers primary purpose is to bring awareness in order to prevent such attacks and maintain privacy and security. The main idea of *A Hybrid Technique For SQL Injection Attacks Detection and Prevention* is to create a hybrid technique that will maintain information safely from SQL injection attacks by combining static and runtime SQL queries, and will therefore protect the data regardless if online or offline based. The main idea of *A Classification of SQL Injection Attacks and Countermeasures* fixates on the existing types of SQL injection attacks providing more insight in order to protect information. As these two share a main topic of SQL injection attacks, in my review I will further review how these two are connected.

As mentioned in the previous paragraph, these two are connected as they are trying to prevent SQL injection attacks. A major takeaway I gained from reviewing these two articles is that data breaching is becoming more and more of a threat. You will hear about social media companies and the ethics of how our data is being sold. Also, recently I have learned that there has been attacks and with high stake threats. Of course it always easy to say why is privacy and security threats even an issue? From *A Classification of SQL Injection Attacks and Countermeasures* I learned that not only are SQL injection attacks one of the most serious threats, but also that researchers and partitioners are only partially aware of all the attack techniques which would make hacking prevention very challenging.

In addition, I discovered there are two key traits of SQLIAs to describe an attack: injection mechanism and attack intent. The following injection mechanisms include: injection through user input, injection through cookies, injection through server variables, and second-order injections. The other type of SQLIAs would be attack intent which would be: identifying injectable parameters, or performing database finger-printing, determining database schema, extracting data, adding or modifying data, performing denial of service, evading detection, bypassing authentication, executing remote commands, and performing privilege escalation.

Further on the hybrid approach there are three different methods to prevent fraudulent attacks. The first is the normal data exchanging strategy where the data is divided into three system tiers: Presentation Tier, Logic Tier, and Storage Tier. The second approach is the suggest approach strategy which is a runtime detection and prevention technique that follows the the typical exchange of queries between the parties, however unlike the first approach there is an extra step toward for a more secure experience, which is the Data-Tier. This preventing any queries from the outside or inside preventing any data temperaments. The final approach is the suggested approach which is composed of different stages to deny any harmful query from seeping into the database. The first stage of this approach is *replicate system databases* which like the name suggests have a duplicate to keep safe from any SQLIAs which will contain a small amount of sample data. The next stage is *creating "database_Behaviors" database* which would entail a separate database, *"database_Behaviors",* which houses all the of the database system queries and the behaviors coupled with these queries under normal circumstances. The fourth stage, *redirect SQL queries*, in this stage the duplicate will go on first while the original copy is stored for later. *Simple SQL syntax checking,* this involves checking all the queries on multiple levels, as well as checking the duplicates. The following stage would be *virtual execution*, so after the SQL syntax has been checked the query will be called on the duplicate database, "Virtual Database" and here as the process is running, it will monitor and trace the behavior of the SQL query. The final stage which is also the most crucial stage is the *SQLIA detection,* thus the importance of the stage is to detect whether the query received is acceptable. In this stage queries are compared to those in the "*database_Behavior*". So these comparisons occur to test the queries and as well as adding new acceptable queries to test against.

One thing I didn't like from the paper, *A Classification of SQL Injection Attacks and Countermeasures,* was the structure of Sections 2-4. I found myself getting confused as to whether it was an SQLIA or an attack intent. As I was first reading this paper, I thought that there were two categories of which an SQLIA would fall under, so an attack would be either a an attack mechanism or an attack intent. However after reviewing these sections again, I found that SQLIA types and the queries used as attack methods used to carry out the attack intent, intended by the attacker. As mentioned before I found that confusing, and cannot confidently say that I still understand this concept correctly. Overall, I agreed with these papers as I am new to this subject, so I found these informative and educational. However, I do question if a hybrid technique would "prevent all types of SQLIAs in different system categories regardless of the system development language or the database engine." Especially after reading that those studying and working to prevent these attacks are "familiar with only a subset of the wide range of techniques available to attackers who are trying to take advantage of SQL injection vulnerabilities".

In conclusion, coupling these two papers as my choice of review, was helpful as they both explained the SQL inject attacks. *A Classification of SQL Injection Attacks and Countermeasures* went into detail of each attack intent and description enabling me to further understand the measures required to prevent such attacks. Both of these papers provided charts and diagrams to help further my understanding of the types of injection attacks: Tautology, Built-In Functions, Logically Incorrect Queries, Union Query, Stored Procedure, Piggy-Backed Queries, Inference, Alternate Encoding, and the Direct Attack. Thus reading *A Hybrid Technique For SQL Injection Attacks Detection and Prevention* I had the knowledge required to understand what goes into a SQLIA and how to prevent these attacks.

References

Jalal Omer Atoum and Amer Jibril Qaralleh Princess Sumaya University for Technology, Amman, Jordan International Journal of Database Management Systems, *A Hybrid Technique For SQL Injection Attacks Detection and Prevention* (IJDMS) Vol.6, No.1, February 2014

William G.J. Halfond, Jeremy Viegas, and Alessandro Orso, College of Computing Georgia Institute of Technology, *A Classification of SQL Injection Attacks and Countermeasures*