A Misuse Pattern for Retrieving Data from a Database Using SQL Injection

Ernest Alder, Richard Bagley, Swati Paghdar, and Dr. Eduardo Fernandez
Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University

Introduction

To design a secure system, we first need to understand possible threats to the system. We need to understand how the specific components of the architecture are compromised or used by an attacker in order to fulfill her objectives. To do this, we apply misuse patterns [1], to describe threats. A **misuse pattern** describes how a misuse is performed from the point of view of the attacker. It defines the environment where the attack is performed, countermeasures to stop it, and it provides forensic information in order to trace the attack once it happens. Misuse patterns are useful for developers because once they determine that a possible attack can happen in the environment, a corresponding misuse pattern will indicate what security mechanisms are needed as countermeasures.

Retrieving Data from a Database using SQL Injection

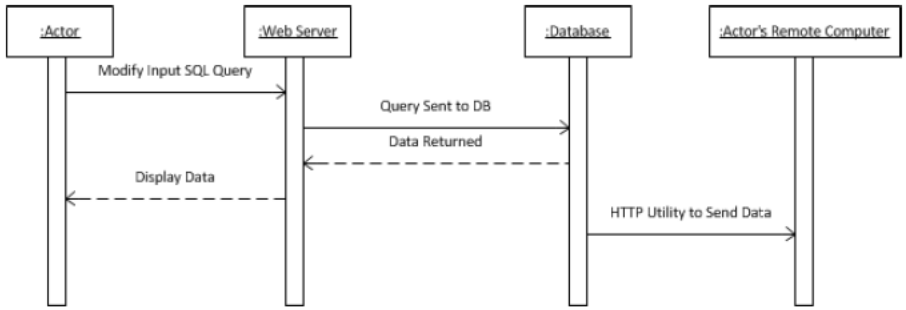
Intent

A SQL injection attack consists of insertion or "injection" of arbitrary code into a SQL by the client in order to alter its intended function, allowing the attacker to retrieve arbitrary amounts of unauthorized data.

Context

Almost all relational databases use SQL interfaces and are thus susceptible to this attack. The database must have remote interfaces accessible through the Internet.

Sequence Diagram for attack



Solution

• Attackers can use SQL manipulation to modify a current SQL query through set operations (e.g. UNION) to return results greater than the original query, that result in illegal access to information.

•The diagram shows how an attack is performed, first assuming the attacker is a legitimate user and then when he is not a legitimate user.

Countermeasures

• Inputs to SQL Strings should be constrained and sanitized in terms of the value that should be provided. For example, an input with an integer as its type should have an integer value passed through it and the information passed from the URL should be limited so that important database information is not imparted to possible hackers.

• Checking for a valid value ranges within the data will also limit SQL Injection because, while a controller of a system might have an idea that a passed value with only have a range of 100 200, an attack would be less likely to know of such a condition.

Forensics

• In most cases, detecting an attack of this kind requires detailed database logging. Through logging of database interaction, an administrator should be able to differentiate legitimate activity from illegitimate access.

• In general, an individual will make multiple tries at breaking into a system. Look for evidence of the attempts of trial and error required to test for flaws in system.

References

1. E.B. Fernandez, N. Yoshioka, and H. Washizaki, "Modeling misuse patterns", 4th Int. Workshop on Dependability Aspects of Data Warehousing and Mining Applications (DAWAM 2009), in conjunction with the 4th Int.Conf. on Availability, Reliability, and Security (ARES 2009). March 16 19, 2009, Fukuoka, Japan

2. OWASP. The Open Web Application Security Project. http://www.owasp.org/index.php/SQL Injection

3. The Business Justification for Data Security. SANS Institute InfoSec Reading Room. http://www.sans.org/reading room/whitepapers /dlp/business justification data security 33033

4. Zetter, Kim. "TJX Hacker Charged With Heartland, Hannaford Breaches." Wired. http://www.wired.com/threatlevel/2009/08/tjx hacker charged with heartland/

5. http://www.integrigy.com/security resources/whitepapers/Integrigy Oracle SQL Injection Attacks.pdf

Undergraduate Research Symposium, April 1, 2011