Event-Driven Attacks on Database Systems

Somendra Chaudhary & Vasanth Pranavan Selvam
Illinois Institute of Technology
Huge Amounts of Data Generated!

Image taken from techcrunch.com
Huge Amounts of Data Generated!

- Huge amount of data generated in today’s world.
- Final destination of this data is the database.
- Database is standard as data can be managed here effectively and efficiently.
- Data manipulation and maintenance done using Database Management System (DBMS).
Need to Secure Data

- Data is very valuable.
- And anything valuable needs to be secured.
- A database immune to attacks (internal or external) is known as a **secure database**.
- **Security attacks** compromise the integrity of database, rendering the system incapable of functioning.
Types of Database Attacks

- SQL (Structured Query Language) injection (SQLi).
- Privilege Abuse.
- Brute-force Attacks.
- Denial of Service (DoS).
- Phishing
- Man-in-the-Middle attacks.
Types of Events Under which Database Attacks are Considered

- Excessive Privileges.
- Weak Audit.
- Many more...
Database is like a Bank Vault!

Image taken from pond5.com
Database is like a Bank Vault!

- Database can be local or outsourced.
- If database is compromised, the very foundation of an organization shakes.
Database is like a Bank Vault and Needs to be Secured

- Attacks may be internal or external.
Database is like a Bank Vault and Needs to be Secured

Do you know what percentage of database attacks are from current or ex-employees?
Database is like a Bank Vault and Needs to be Secured

80%
Data can be at risk when in use, in transit or at rest!
Seuring Stored Procedures

- Organizations use stored procedures.
- Saved with a relegated in an RDBMS.
- Can be shared and reused by different projects.
- Two phase technique discussed by Som, Sinha and Kataria to secure stored procedures:
  - If first phase fails to secure, then second phase is invoked!
Kellaris, Kollios, Nissim and O’Neil have proposed abstract models. These models capture secure outsourced storage systems in sufficient generality. They identify two basic sources of leakage, namely access pattern and communication volume.
Malik and Patel highlighted many possible attacks.
Also proposed counter measures.
Finally proposed possible control methods.
SQL Injection: Example 1: External

http://students.com?studentId=117 or 1=1;--

SELECT * FROM students WHERE studentId=117 or 1=1;

Attacker

Data for all students is returned to the attacker

Web API Server

Return data for all students

SQL Database Server

Image taken from cloudflare.com
SQL Injection: Example 2: Internal

```
' UNION SELECT username, password FROM users--
```

```
SELECT name, description FROM products WHERE category = 'Gifts' UNION SELECT username, password FROM users--
```
SQL Injection

- Type of Injection attack.
- SQL commands are injected into the input of the data plane to execute pre-defined SQL commands.
- Uses malicious SQL queries to access information not intended for display.
- Information may include sensitive company data, user lists or details of private customers.
SQL Injection has far reaching negative affects

- Unauthorized viewing of user lists.
- Removing whole tables.
- Gaining administrative rights to a database from the attacker.
- Loss of customer trust if phone numbers, addresses and credit card details are stolen.
SQL Injection usually targets websites!

Welcome Back!

Login With Your Username And Password

' or 1=1#

...
SQL Injection usually targets websites!

- SQL queries are arbitrarily inserted into web-based app database by an unauthorized user.
- Basic form of SQL injection is the user input.
- Generally forms are used for user to input the data for the respective field.
- Through this form the data entered in the front-end is passed to the backend database.
SQL Injection usually targets websites!

- SQL queries are arbitrarily inserted into web-based app database by an unauthorized user.
- Basic form of SQL injection is the user input.
- Generally forms are used for user to input the data for the respective field.
- Through this form the data entered in the front-end is passed to the backend database.
SQL Injection usually targets websites!

- The input entered from the form by the user is rendered invalid by the web app.
- An attacker will be able to inject an SQL into the backend database
SQL Injection: After system is compromised

● The attacker may:
  > Take copy of the content.
  > Modify content.
  > Remove content: entire tables!!
  > Poison cookies used to locally store information of customers.

● HTTP headers can be used as an attack by injecting code into the backend database *if the inputs are not sanitized by the web application.*
EVERYONE IS ENTITLED TO BE STUPID
BUT YOU ABUSE THE PRIVILEGE

Image taken from me.me
Privilege Abuse

- Clients (or applications) may be conceded database privileges that surpass the prerequisites of their activity work.
- These benefits might be utilized to access private data.
Privilege Abuse: Consider a college director

Image taken from stc-ea.com
Privilege Abuse: Consider a college director

● The activity of the director requires just read-only access.
● Consider if they are given unreasonable update benefits.
● Then there is a chance that they might use these benefits to update to alter grades of students!
What’s the solution to this problem?
For one, have a good hiring policy!

“This is the worst résumé I’ve ever seen!”
Privilege Abuse: Solution

- Query level access control.
- Confines privileges to least required activities and information.
- Most databases provide abilities like: RLS, triggers.
- The manual plan of these devices make them illogical in everything except the most restricted organizations.
Privilege Abuse

- Clients may manhandle genuine information get to benefits for unapproved purposes.
- A client may have benefits to see singular patient records by means of a custom social insurance application.
- The client may manhandle that benefit to recover every single patient record through a MS-Excel client!
The idea is access control strategies that apply not exclusively to what information is available, however how information is accessed.
Privelege Abuse: Solution

By implementing strategies for time of day, area, and application customer and volume of information recovered, it is conceivable to distinguish clients who are mishandling access benefits.
Privelege Abuse

- A programmer may exploit vulnerabilities in database to change over low-level access rights to abnormal access privileges.
- Privilege escalation attacks can be vanquished with a blend of query level access control and conventional intrusion prevention system (IPS).
Privilege Abuse

- Query level access control can distinguish a client who abruptly utilizes an uncommon SQL activity,
- While an IPS can recognize a particular archived danger inside the task.
- In most database establishments, the Least Privilege Principle isn't clung to: but why?
Privilege Abuse: Why isn’t Least Privilege Principle Adhered To?

Image taken from imgflip.com
Privilege Abuse: Why isn’t Least Privilege Principle Adhered To?

- The programming staff probably don't know any better.
- They may improve yet figure they don't have sufficient resources to actualize this accurately.
- The organizations simply don’t believe it to be as big a risk that it actually is.
Brute Force Attack

Image taken from makeuseof.com
What is a Brute Force Attack?

- A Brute Force Attack is the simplest method to gain access to a site or server (or anything that is password protected).
- It tries various combinations of usernames and passwords until it gets in.
- This repetitive action is like an army attacking a fort.
Brute Force Attack: How it is done?

- Every common ID (for e.g. “admin”) has a password.
- All you need to do is try to guess the password. Simple, isn’t it?
- Well, not really!
Brute Force Attack: How it is done?

- Let’s say if it’s a 4-digit-pin, you have 10 numeric digits from 0 to 9. This means there are 100 possibilities.
- You can figure this out with pen and paper like Mr. Bean did in the movie, Mr. Bean’s Holiday.
- But, the truth is that no password in the world consists of only 4 characters.
## Results and Findings

<table>
<thead>
<tr>
<th></th>
<th>Excessive Privileges</th>
<th>Weak Audit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLi</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Privilege Abuse</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Brute-force Attacks</td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>

**Key**

<table>
<thead>
<tr>
<th>X-axis</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y-axis</td>
<td>Attack</td>
</tr>
</tbody>
</table>
Conclusion

1. In order to prevent against attacks viz., SQLI and Privilege abuse, while granting privileges, it must be strictly adhered that to that excessive privileges are NOT granted!

2. A strong audit shall ensure protection against SQLI, Privilege Abuse and Brute-force attacks.
Future Work
Future Work

The processes described in our paper have not been practically implemented: so practically implement them to verify our findings.
More types of attacks such as Denial of Service, Phishing, Man-in-the-Middle (MITM) attacks and et cetera may be taken into consideration.
We have considered two events in this paper. Many more events may be taken into consideration!!
Thank You