Workload Modeling for Security and Privacy in Databases

Gokhan Kul, Ph.D.
Assistant Professor
Outline

- Insider Threat Overview
- Workload Modeling
- PocketData Project
- Insider Threats Project
- Other Projects
- Background
In a standard office environment, there are strong defense mechanisms:

- Firewall
- Password protection
- Antimalware
- VPN Encryption
- IDS
- DLP
- Many more…
Your office
OMG! What did I do? Did I just send $5000 to the wrong account?

If they don’t pay me what I’m worth, I know how I can take it.
Any information system of the organization
Outline

- Insider Threat Overview
- Workload Modeling
- PocketData Project
- Insider Threats Project
- Other Projects
- Background
Traditional Workload Modeling

Question Asked:

What kind of queries do we receive?
Traditional Workload Modeling

Question Asked:
What should we focus on to increase performance?

Database Structure

Primary Keys
Foreign Keys
Indexes
Joins
Application: Benchmarks

Measure Throughput & Latency

Latency:

is the time required to perform one single action

Throughput:

is the number of such actions executed or results produced per unit of time
Application: Benchmarks

Which one is more important at database performance?

Latency vs Throughput

Hold that thought
Improvement Points

No attention to the activity performed
SELECT on a table with 10 rows vs. 1.000.000 rows
1 access attempt to a row vs 1.000 access attempt

No attention to what the user intends to do
Bring me a customer who’s a frequent customer
vs bring me a customer who last shopped last week
Outline

- Insider Threat Overview
- Workload Modeling
- PocketData Project
- Insider Threats Project
- Other Projects
- Background
PocketData: Databases on Smartphones

Databases Are Single Client
Latency, Not Throughput, Matters

Workloads Are Bursty

Representative Benchmarks Matter
With Great Differences Come Great Opportunities
Be Smart and Lose the I: ACID => ACD

The Cost of Database Isolation

Can we design databases with weaker ACID (more Basic) semantics?

Atomicity, Consistency, Isolation, Durability
Optimize for Burst Response

Some observed throughputs:
Optimize for Burst Response

Some observed throughputs:

36,000 tpm* ~ 600 tps*  
112,000 tpm*

*Oliver Kennedy, Jerry Antony Ajay, Geoffrey Challen, and Lukasz Ziarek. 2015. Pocket Data: The Need for TPC-MOBILE. In TPC-TC.

*http://www.tpc.org/tpcc/results/tpcc_results.asp
Optimize for Burst Response

A typical database operation pattern on a mobile device:

Since we don’t have to worry about throughput,
How much can we improve latency?
Security Implications

A typical database operation pattern on a mobile device:

How does a burst change for each user? Can we distinguish different users? Is it possible to perform a side channel attack? Can defense mechanisms respect privacy?
PocketData: Experiments Performed

Two Phases:

(1) 11 lab members

(2) 56 phones deployed in the wild

Next
PocketData: Grant Proposal

NSF CISE Community Research Infrastructure (CCRI) (January 2022): Let’s create a stable testbed and distribute these phones to a larger group (New data servers, new software versions, etc)
PocketData: Initial Results

Procedures trigger sequence of queries:

First few queries of a burst helps predicting the rest of the queries

Behavior patterns can distinguish users from each other
Procedure latency:
How updates in the software will affect the procedure latency?
Should I push this update or not to the software?
New, Representative Benchmarks

A typical database comparison study:

But scaling doesn’t matter on phones.
New, Representative Benchmarks

Databases are per-app

The corner case is the common case
Contributers

Gokhan Kul (Umass Dartmouth)
Gourab Mitra (Datometry)
Carl Nuessle (UB)
Darshana Balakrishnan (UB)
Lukasz Ziarek (UB)
Oliver Kennedy (UB)

Interested Community

Arnab Nandi (Ohio State University)
Richard Hipp (SQLite)
Stratos Idreos (Harvard University)
More...
Outline

- Workload Modeling
- PocketData Project
- Insider Threats Project
- Other Projects
- Background
Insider Threat

A trusted person (TP): Employee, Contractors, Vendors

TP may misuse legitimate access:
- Unintentional – incompetency, amateur behavior
- Intentional – Traitor
- Collusion

TP may obtain unauthorized access:
- Masquerading
The Problem

The attackers **know** you and you **trust** them

They are inside (almost) all of the security layers
Data Access Architecture
Data Access Architecture
Data Access Architecture
Data Access Architecture
Method

Improvement Points

How to make anomaly detection better?
(1) Find ideal similarity metrics for query clustering
(2) Standardize (called Regularization) queries
(3) Exploit user’s distinct behavior
(4) Exploit changes in user’s habits
Improvement Point (1) & (2)

Gokhan Kul, Duc Luong, Ting Xie, Varun Chandola, Oliver Kennedy, and Shambhu Upadhyaya. *Similarity Metrics for SQL Query Clustering*, IEEE Transactions on Knowledge and Data Engineering (TKDE), 2018.
Improvement Point (3)

Can we distinguish two users based on their activity patterns?

Google+ application, 2M SQL queries, 11 users, 1 month

KL-Divergence score heat map for 11 Google+ users
Improvement Point (4)

Can we profile a user based on changing habits?

Google+ application, 2M SQL queries, 11 users, 1 month

Behavior change based on SQL Queries for 11 Google+ users
# Data from PocketData

<table>
<thead>
<tr>
<th>Application</th>
<th># of Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Dataset</td>
<td>45,090,798</td>
</tr>
<tr>
<td>Facebook</td>
<td>1,212,779</td>
</tr>
<tr>
<td>Google+</td>
<td>2,040,793</td>
</tr>
<tr>
<td>Hangouts</td>
<td>974,349</td>
</tr>
<tr>
<td>Google Play Services</td>
<td>14,813,949</td>
</tr>
<tr>
<td>Media Storage</td>
<td>13,592,982</td>
</tr>
</tbody>
</table>
Simulated Attacks (Queries written by us)

<table>
<thead>
<tr>
<th>Service</th>
<th># of Attacks Performed</th>
<th># of Attacks Detected</th>
<th>Success</th>
<th># of Attacks Detected</th>
<th>Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>105</td>
<td>97</td>
<td>92.4%</td>
<td>98</td>
<td>93.3%</td>
</tr>
<tr>
<td>Google+</td>
<td>225</td>
<td>202</td>
<td>89.8%</td>
<td>214</td>
<td>95.1%</td>
</tr>
<tr>
<td>Hangouts</td>
<td>239</td>
<td>206</td>
<td>86.2%</td>
<td>206</td>
<td>86.2%</td>
</tr>
<tr>
<td>Google Play</td>
<td>282</td>
<td>261</td>
<td>92.6%</td>
<td>267</td>
<td>94.7%</td>
</tr>
<tr>
<td>Media Storage</td>
<td>282</td>
<td>251</td>
<td>89.0%</td>
<td>259</td>
<td>91.8%</td>
</tr>
</tbody>
</table>
# Real Workload Attacks
(Queries injected from other users)

<table>
<thead>
<tr>
<th>Service</th>
<th># of Attacks Performed</th>
<th>Ideal Threshold</th>
<th>Behavior Drift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Detected</td>
<td>Success</td>
</tr>
<tr>
<td>Facebook</td>
<td>315</td>
<td>290</td>
<td>92.1%</td>
</tr>
<tr>
<td>Google+</td>
<td>2025</td>
<td>1817</td>
<td>89.7%</td>
</tr>
<tr>
<td>Hangouts</td>
<td>2201</td>
<td>1842</td>
<td>83.7%</td>
</tr>
<tr>
<td>Google Play</td>
<td>2583</td>
<td>2066</td>
<td>80.0%</td>
</tr>
<tr>
<td>Media Storage</td>
<td>2583</td>
<td>2099</td>
<td>81.3%</td>
</tr>
</tbody>
</table>
Contributers

Gokhan Kul (Umass Dartmouth)
Shambhu Upadhyaya (UB)
Varun Chandola (UB)
Oliver Kennedy (UB)
Ting Xie (UB)
Duc Luong (UB)
Long Nyugen (UMich)
Outline

- Workload Modeling
- PocketData Project
- Insider Threats Project
- Background
Background

Research
Cybersecurity of Database & Cloud Systems