UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

CS411 - Advanced Relational Databases

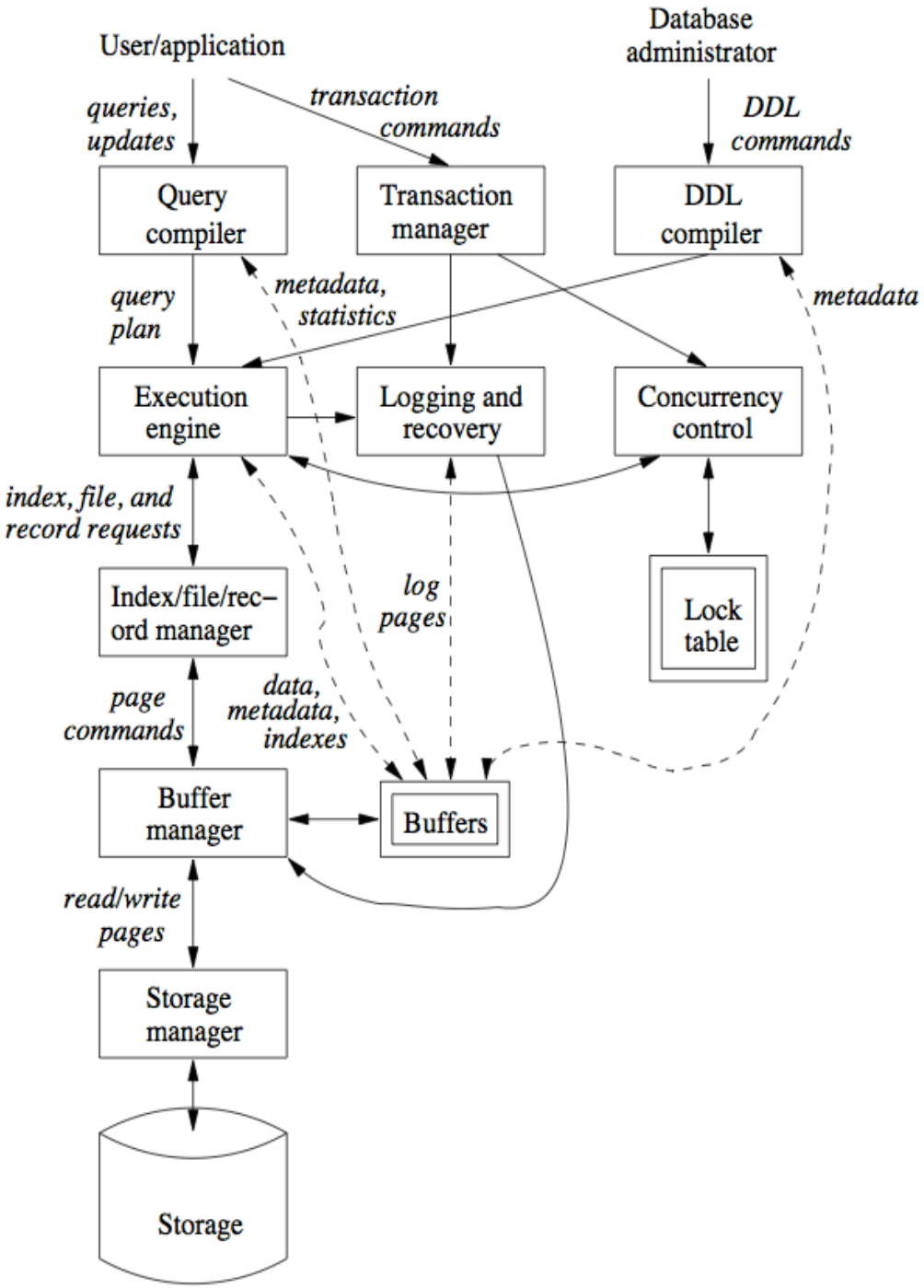
illinois.edu

Announcements

- Last call for conflicts…
- MP3 due tonight
- Project Stage 5 due next week
  - Schedule a demo
  - Have a video and final report finished
  - Hand in .zip of your code (can use svn)

Announcements

- Bad news:
  - HW4 will come out next Wednesday
- Good news:
  - It will be easy
  - Dropping the lowest HW/MP score



illinois.edu


UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN
1337 H@X d00d
illinois.edu

The News-Gazette
SERVING EAST CENTRAL ILLINOIS

Paid with plastic at Schnucks? Your info is at risk

Mon, 04/15/2013 - 9:00pm | The Associated Press

ST. LOUIS — Up to 2.4 million credit cards and debit cards used by customers at Schnucks grocery stores in four states may have been compromised over a three-month period, the suburban St. Louis chain said Monday.

Other Related Content

- Police advise getting new card numbers

Customers of the stores in Champaign, Urbana and Savoy are among the people potentially affected.

Schnucks Markets Inc. for the first time outlined the potential breadth of the fraud that came to light last month. Many customers have reported fraudulent charges, some in the thousands of dollars.

Schnucks

STORE LOCATIONS

ABOUT SCHNUCKS | DEPARTMENTS | SERVICES | HEALTH & WELLNESS | FAMILY & FUN | WEEKLY AD

Schnucks Releases Details of Card Issue as Investigation Nears End

See Potentially Affected Stores, Timeline and Message from Scott Schnuck

Click to Watch Video

Latest Communication

- SCHNUCKS RELEASES DETAILS OF CARD ISSUE AS INVESTIGATION NEARS END
- SCHNUCKS PROVIDES ADDITIONAL INFORMATION ON CREDIT CARD ISSUE
- SCHNUCKS ANNOUNCES CREDIT CARD ISSUE FOUND AND CONTAINED

Departments

Floral
We're committed to providing an extensive selection of roses, fresh flowers, green and blooming plants, balloons and gourmet baskets for every occasion.

Pharmacy
Refill your prescription online, learn more about our pharmacy programs and get Health Information for You and Your Family.
See our Pharmacy Page.

Bakery
Browse through our most popular products, view our decorated cake album and plan your next party!

Boar's Head
Find a Schnucks location near you offering Boar's Head products.

Fish Frydays
Check out our Fish Frydays!

Weekly Ad
Find great deals in your area now!

Locations
Find a Schnucks store near you!

Newsletter
Sign up today to receive exciting offers!

The Schnucks Difference

facebook | YouTube | Twitter

OCTOBER 3, 2012, 8:34 PM | 8 Comments

Hackers Breach 53 Universities and Dump Thousands of Personal Records Online

By NICOLE PERLROTH

Hackers published online Monday thousands of personal records from 53 universities, including Harvard, Stanford, Cornell, Princeton, Johns Hopkins, the University of Zurich and other universities around the world.

The group of hackers, calling themselves Team GhostShell, claimed responsibility for the attack on Twitter and published some 36,000 e-mail addresses and thousands of names, usernames, passwords, addresses and phone numbers of students, faculty and staff, to the Web site Pastebin.com. In most cases the data was already publicly available, but in some instances the records included additional sensitive information such as students' dates of birth and payroll information for university employees.

Typically, hackers seek such information because it can be used to steal identities, crack bank accounts or can be sold on the black market. Universities make ripe targets because they store vast numbers of personal records, often in decentralized servers. The records can be a gold mine because students often have pristine credit reputations and do not monitor their account activity and credit scores as vigilantly as adults.

Data Breaches

- PSN data breach leaked 70 million user's credit card data
- Zappos leaked 24 million names, email addresses, phone numbers, addresses
- DoD, DoE, NASA, DHS, etc.
- etc. etc. etc.

How does this happen?

- 80% of attacks are against web applications - Verizon

Demo 1

userId	username	password	email
1	temp_user	p@$$w0rd	user@user.com
2	temp_user2	monkey	hello@hello.com
3	temp_user3	tempuser3	what@aol.com
4	temp_user4	robby121	robby121@yahoo.com

SQL Injection

- 97% due to SQL injection
- SQL injection is still the #1 attack vector
  - FireHost, Imperva
- Lulzsec got a million plaintext usernames and passwords from Sony purely through SQL injection

Demo 2

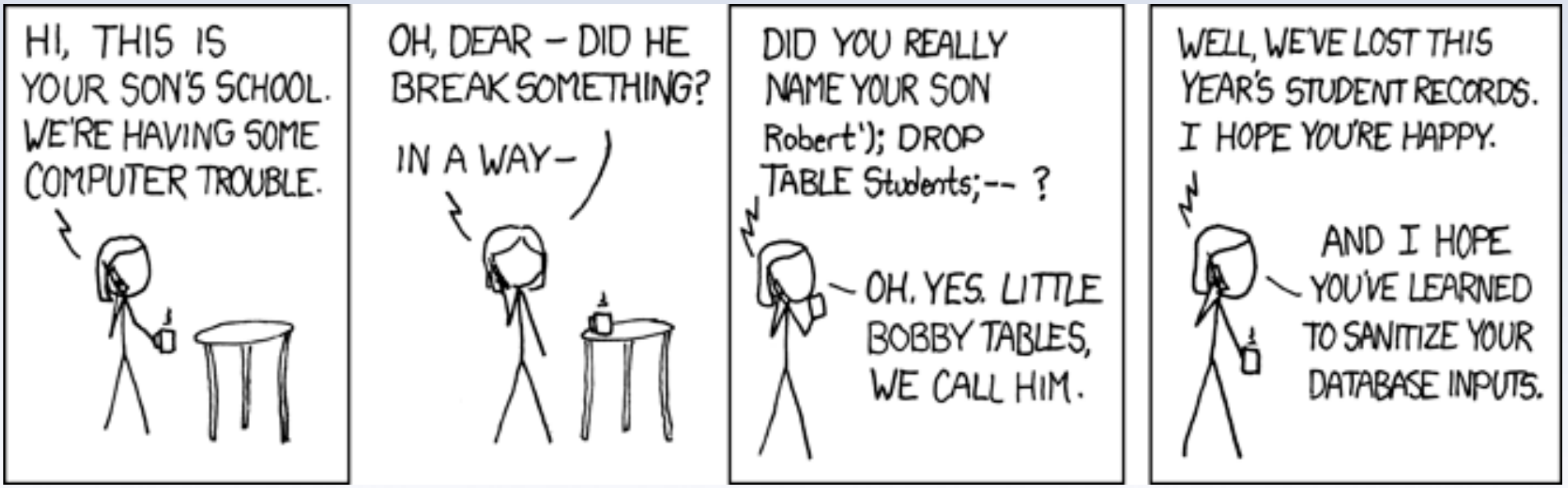
- Advanced attacks leverage the web application to attack the server
  - Use SQL Injection to gain access to database/ accounts
  - Use database vulnerabilities to access server
  - Escalate privileges, install back door/rootkit
  - Work deeper into the network

SQL Injection

- In-band attacks
  - sometimes, query results are not displayed
  - use SQL error messages to infer query results
- Blind SQL injection
  - No error messages returned

Demo 3

- Blind SQL injection
- Entire Navy/DHS user database compromised this way!


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Mitigation

- Attackers will always go after the weakest system
- Implement layers of security
- If one layer is breached, users are still safe
- What failed and how can we stop it?

1. Properly escape user input

- Put back slashes before special characters
- NOT on the client side
  - ***NEVER EVER EVER*** validate/sanitize user input with Javascript
- php has helper functions for this mysql_real_escape_string();

2. Validate user input

- Use regular expressions
  - Are usernames allowed to have spaces, semicolons, quotes, etc?
  - No, you say?
  - Then why let those go to the database?!

3. Prepared statements

- SQL statements are precompiled
  - user variables are filled in later by application
  - ONLY interpreted as values by DBMS
  - Added advantage: more efficient
- Example (PHP)

```
$stmt = $dbh->prepare("SELECT * FROM users
WHERE USERNAME = ? AND PASSWORD = ?");
$stmt->execute(array($username, $password));
```

Good enough?

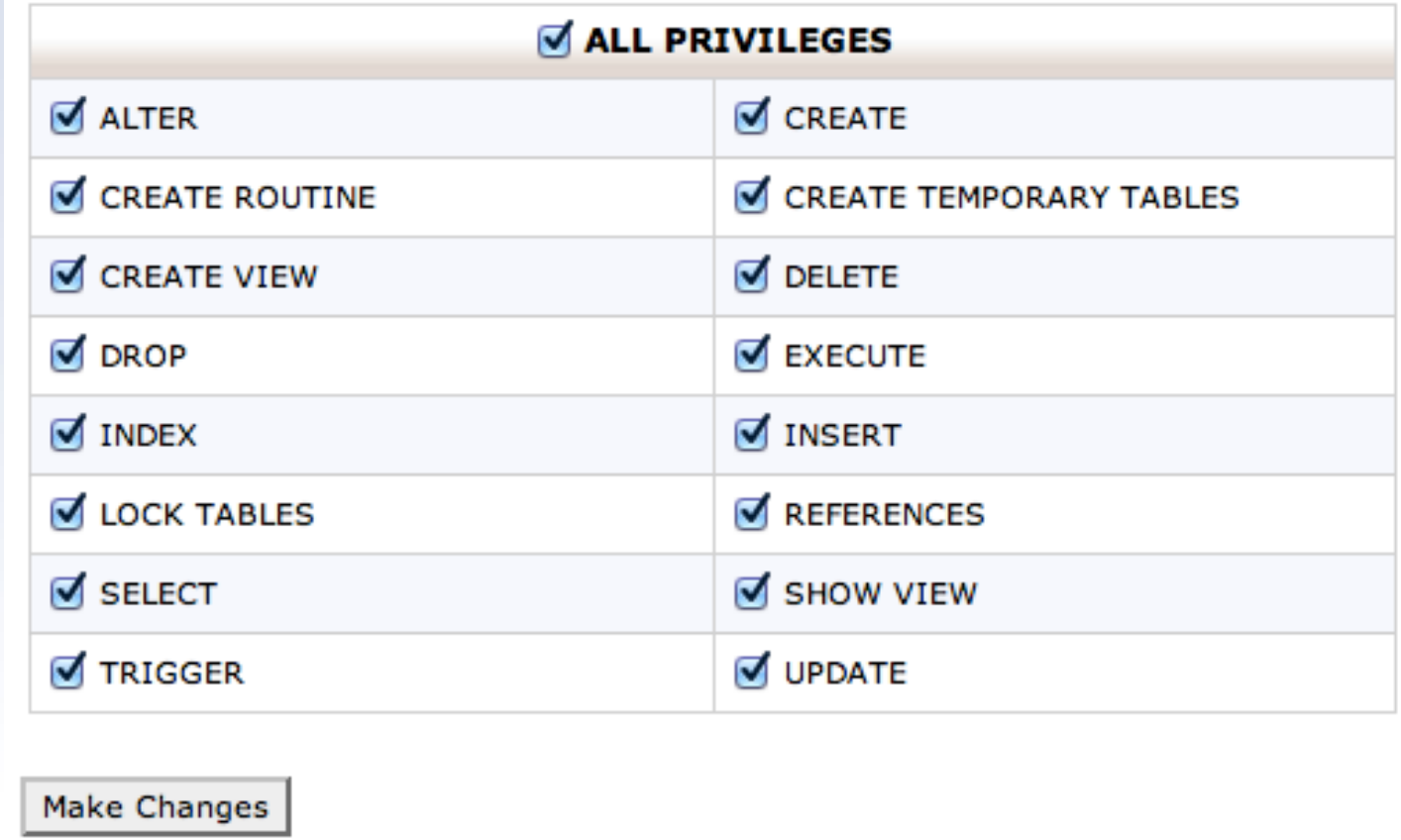
- NO!
  - Other vulnerabilities could exist
  - Could be leveraged to alter prepared statement string
  - Server software could have vulnerabilities (Apache vulnerabilities?! NEVER!)
  - Plugins could introduce vulnerabilities

4. Secure the database

- Users are granted privileges on SQL database elements
  - Databases, tables, etc.
- Users with GRANT privilege can in turn grant privileges to other users
- Can also REVOKE privileges from a user

ALL PRIVILEGES	
☐ ALTER	☐ CREATE
☐ CREATE ROUTINE	☐ CREATE TEMPORARY TABLES
☐ CREATE VIEW	☐ DELETE
☐ DROP	☐ EXECUTE
☐ INDEX	☐ INSERT
☐ LOCK TABLES	☐ REFERENCES
☐ SELECT	☐ SHOW VIEW
☐ TRIGGER	☐ UPDATE

Make Changes


ALL PRIVILEGES
ALTER
CREATE
CREATE ROUTINE
CREATE TEMPORARY TABLES
CREATE VIEW
DELETE
DROP
EXECUTE
INDEX
INSERT
LOCK TABLES
REFERENCES
SELECT
SHOW VIEW
TRIGGER
UPDATE
Make Changes

4. Secure the database

- Only give users privileges they need
  - Web application user should NOT have DELETE, CREATE, DROP, UPDATE privileges
- Create admin and application users with appropriate privileges

Good enough?

- NO!
  - Database software itself could have vulnerability
  - Plugins could introduce vulnerabilities
  - Hard drives/backups are lost and stolen

5. Hash passwords

- Don’t store passwords in plain text
  - EVER
  - This should really never happen
  - Seriously. Never.
  - If you let a company you work for do this, you have failed your customers, your company, and all of us

5. Hash Passwords

- Trivially easy in php
  - sha1('mypassword') returns b467b644150eb350bbc1c8b44b21b08af99268aa
- Reversing cryptographic hash function is not possible
- Even with the hashed password, attacker has to supply the password for the hash

Good enough?

- NOOOOOOOOO!
  - Rainbow tables exist all over the web
    - Hash common passwords and store all the results
    - Look up user's password hash, recover password
  - Offline password crackers all over the web
    - Just keep hashing passwords until you get a match
    - e.g. John the Ripper - multicore/cluster support

5.1 Salt ***and*** hash passwords

- Create an extra bit of random data for each user called “salt”
- Hash this with the password
- Resulting hash function is unique for each user

5.2 When possible…

- Popular hash functions are very fast
  - usually implemented so they can take advantage of hardware
  - this helps attacker, but not us
- To slow down attackers, use a slow or memory-hard hash function
  - e.g. bcrypt

Good enough?

- Do I even have to answer this anymore?

6. Encrypt the database

- All sensitive data stored in the database should be encrypted
- Disk itself should be encrypted

And…

- Admin usernames and passwords should not be default values
- Web server should be correctly configured
- etc. etc. etc.