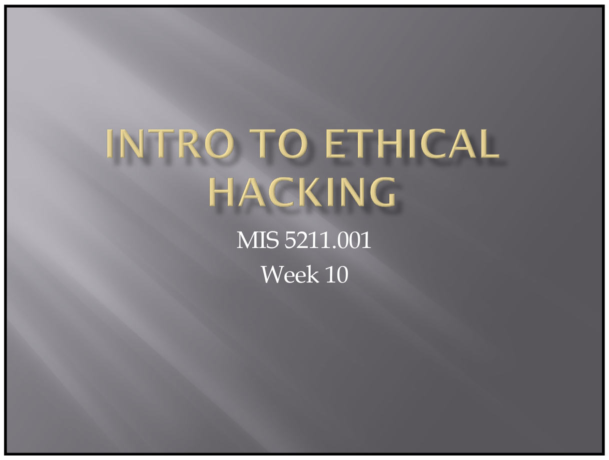


INTRO TO ETHICAL HACKING

MIS 5211.001

Week 10

1

Tonight's Plan

- More Web Application Security

MIS 5211.001 2

2

Recall

- Download SecurityShepherd
  - https://github.com/OWASP/SecurityShepherd/releases
- Download Security Dojo
  - https://sourceforge.net/projects/websecuritydojo/

MIS 5211.001 3

3

Cross Site Scripting (XSS)

- Malicious JavaScript code inserted (injected) into a web site or page

MIS 5211.001 4

4

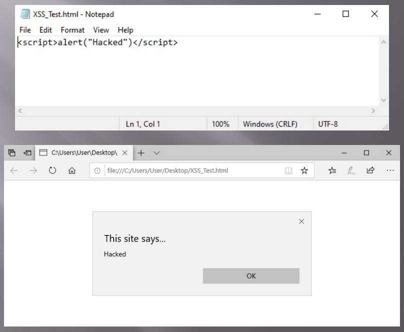
Types of XSS

- 3 Types
  - Persistent (Stored) XSS
  - Non-Persistent (Reflected) XSS
  - DOM based (Document Object Model) XSS
- YouTube Video
  - XSS - Cross Site Scripting Explained – YouTube
  - https://www.youtube.com/watch?v=IuzU4y-UjLw

MIS 5211.001 5

5

Basic Example


XSS_Test.html - Notepad
File Edit Format View Help
C:\Users\User\Desktop\
file:///C:/Users/User/Desktop/XSS_Test.html
OK

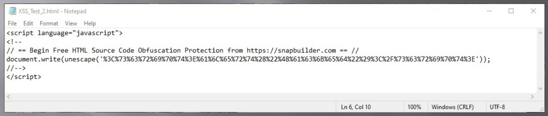
```
<script>alert("Hacked")</script>
```

```
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

```
This site says...
Hacked
```

6

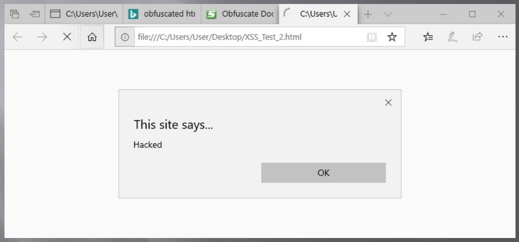
New Example



```
<script language="javascript">
<!--
// == Begin Free HTML Source Code Obfuscation Protection from https://snapbuilder.com == //
document.write(unescape('%3C%73%63%72%69%70%74%3E%61%6C%65%72%74%28%22%48%61%63%6B%65%64%22%29%3C%2F%73%63%72%69%70%74%3E'));
//-->
</script>
```

7

New Result



This site says...

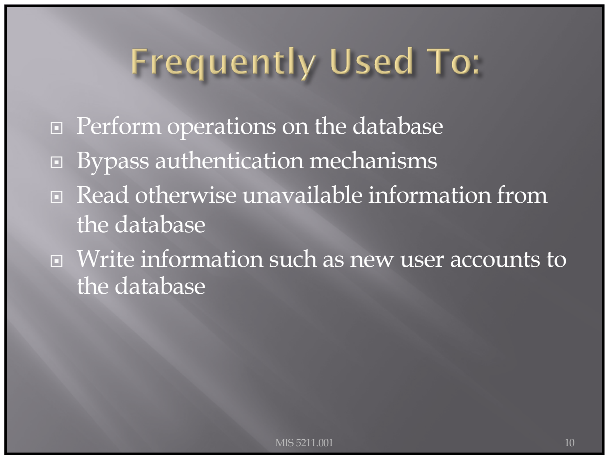
Hacked

8

SQL Injection

- We are going to cover some "Basics"
- SQL Injection is a subset of the general flaw "Injection" covered last week
- Client supplied data passed to an application without appropriate data validation
- Processed as commands by the database
- Remember in all of this that we can also use the intercepting proxy to "add" text the browser doesn't want to accept

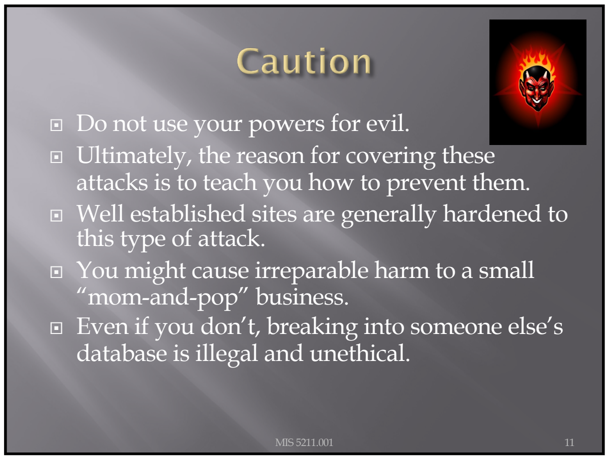
9



Frequently Used To:

- Perform operations on the database
- Bypass authentication mechanisms
- Read otherwise unavailable information from the database
- Write information such as new user accounts to the database

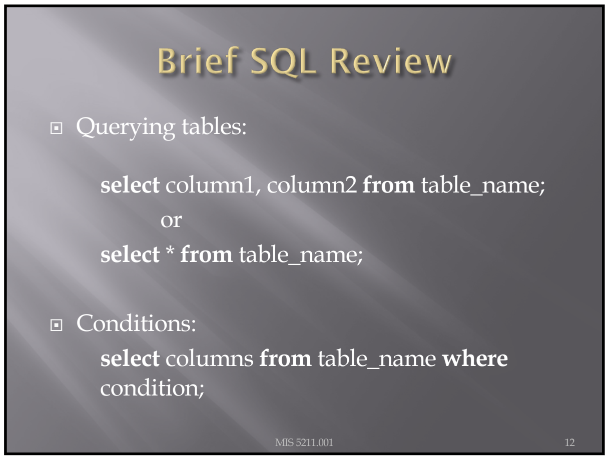
10



Caution

- Do not use your powers for evil.
- Ultimately, the reason for covering these attacks is to teach you how to prevent them.
- Well established sites are generally hardened to this type of attack.
- You might cause irreparable harm to a small "mom-and-pop" business.
- Even if you don't, breaking into someone else's database is illegal and unethical.

11



Brief SQL Review

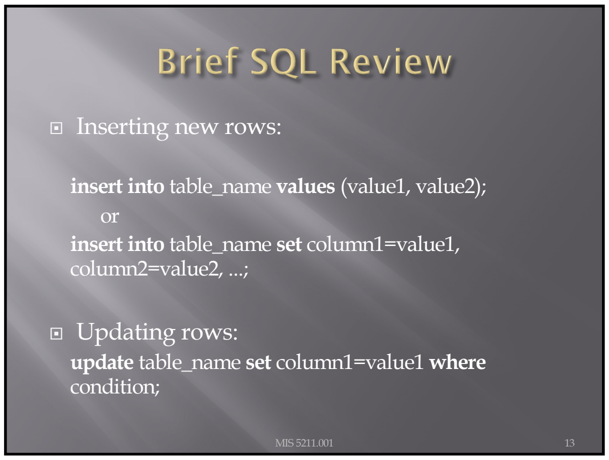
- Querying tables:

  select column1, column2 **from** table_name;
  or
  select * **from** table_name;

- Conditions:

  select columns **from** table_name **where** condition;

12



Brief SQL Review

- Inserting new rows:

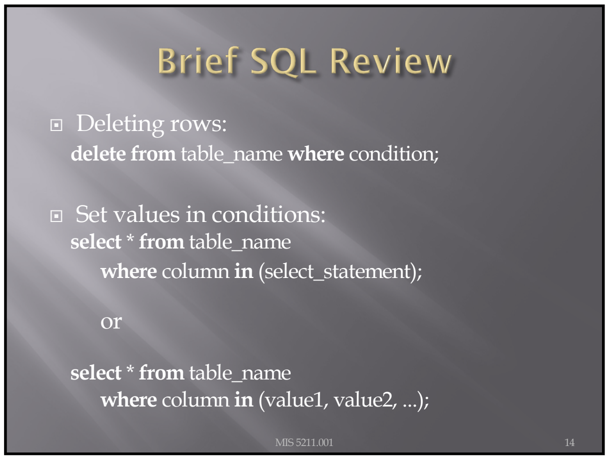
insert into table_name **values** (value1, value2);
or
insert into table_name **set** column1=value1, column2=value2, …;

- Updating rows:

update table_name **set** column1=value1 **where** condition;

MIS 5211.001 13

13



Brief SQL Review

- Deleting rows:

delete from table_name **where** condition;

- Set values in conditions:

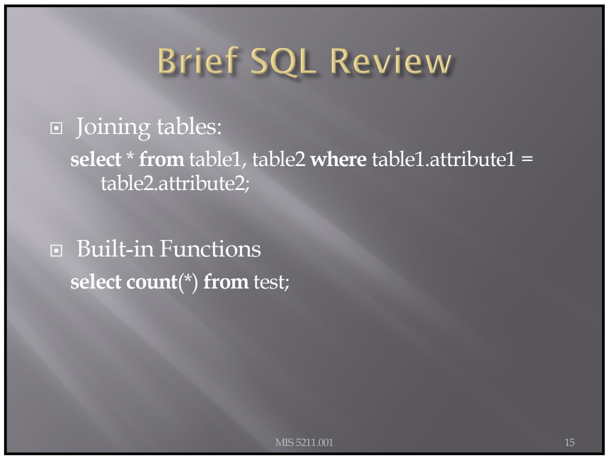
```
select * from table_name
    where column in (select_statement);

    or

select * from table_name
    where column in (value1, value2, …);
```

MIS 5211.001 14

14



Brief SQL Review

- Joining tables:

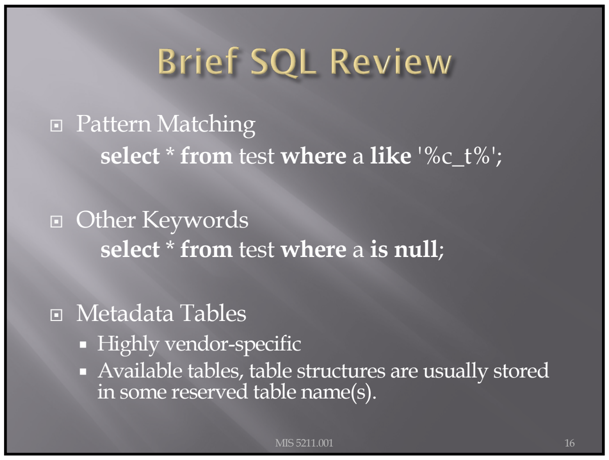
select * **from** table1, table2 **where** table1.attribute1 = table2.attribute2;

- Built-in Functions

select count(*) **from** test;

MIS 5211.001 15

15

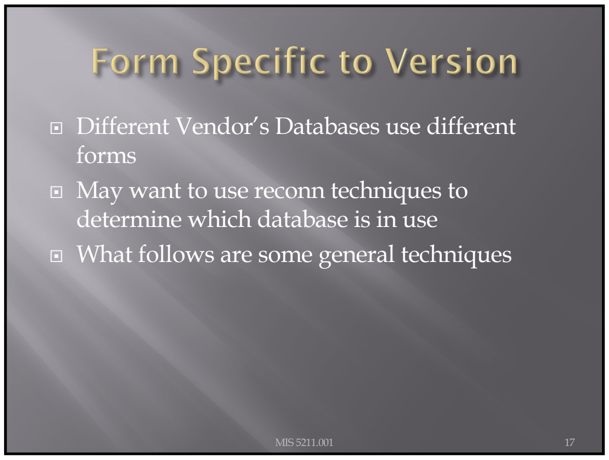


Brief SQL Review

- Pattern Matching
  select * **from** test **where** a **like** '%c_t%';
- Other Keywords
  select * **from** test **where** a **is null**;
- Metadata Tables
  - Highly vendor-specific
  - Available tables, table structures are usually stored in some reserved table name(s).

MIS 5211.001 16

16

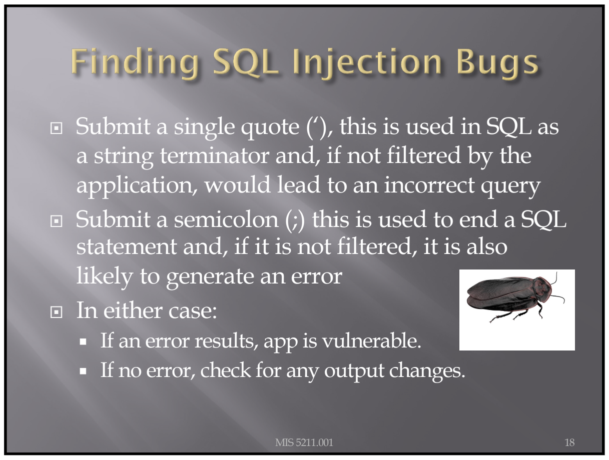


Form Specific to Version

- Different Vendor's Databases use different forms
- May want to use reconn techniques to determine which database is in use
- What follows are some general techniques

MIS 5211.001 17

17

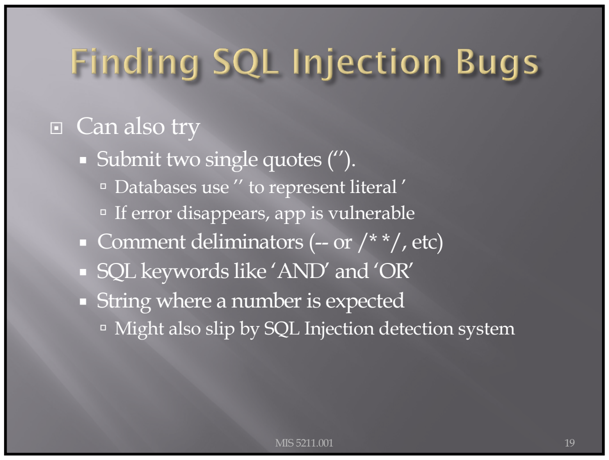


Finding SQL Injection Bugs

- Submit a single quote ('), this is used in SQL as a string terminator and, if not filtered by the application, would lead to an incorrect query
- Submit a semicolon (;) this is used to end a SQL statement and, if it is not filtered, it is also likely to generate an error
- In either case:
  - If an error results, app is vulnerable.
  - If no error, check for any output changes.

MIS 5211.001 18

18

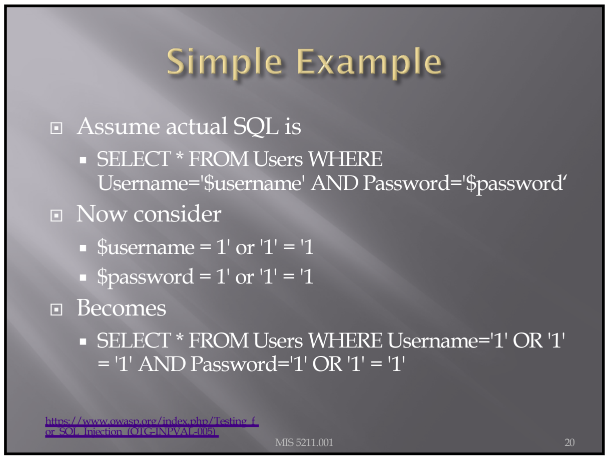


Finding SQL Injection Bugs

- Can also try
  - Submit two single quotes (’’).
    - Databases use ’’ to represent literal ’
    - If error disappears, app is vulnerable
  - Comment deliminators (-- or /* */, etc)
  - SQL keywords like ‘AND’ and ‘OR’
  - String where a number is expected
    - Might also slip by SQL Injection detection system

MIS 5211.001 19

19



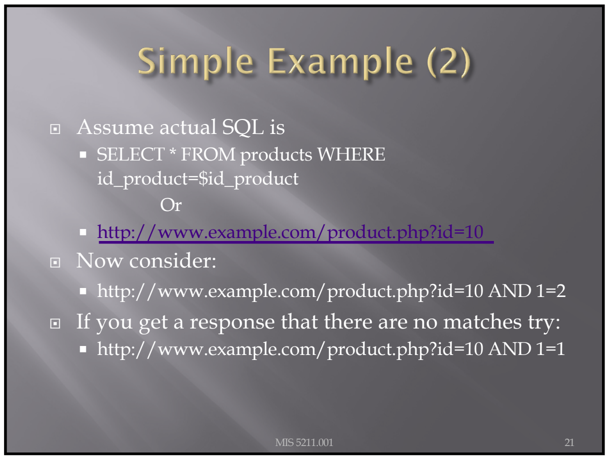
Simple Example

- Assume actual SQL is
  - SELECT * FROM Users WHERE Username='$username' AND Password='$password‘
- Now consider
  - $username = 1' or '1' = '1
  - $password = 1' or '1' = '1
- Becomes
  - SELECT * FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'

https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)

MIS 5211.001 20

20



Simple Example (2)

- Assume actual SQL is
  - SELECT * FROM products WHERE id_product=$id_product

    Or
  - http://www.example.com/product.php?id=10
- Now consider:
  - http://www.example.com/product.php?id=10 AND 1=2
- If you get a response that there are no matches try:
  - http://www.example.com/product.php?id=10 AND 1=1

MIS 5211.001 21

21

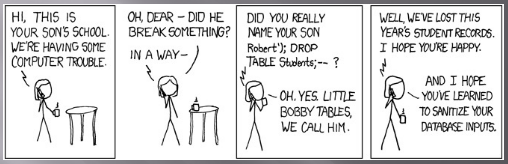
Fingerprinting Databases

- Look at your error messages
- MySQL
  - You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '\'' at line 1
- SQL Server
  - ORA-00933: SQL command not properly ended
- PostgresSQL
  - Query failed: ERROR: syntax error at or near "′" at character 56 in /www/site/test.php on line 121.

MIS 5211.001 22

22

Famous SQL Humor


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.

http://xkcd.com/327/

MIS 5211.001 23

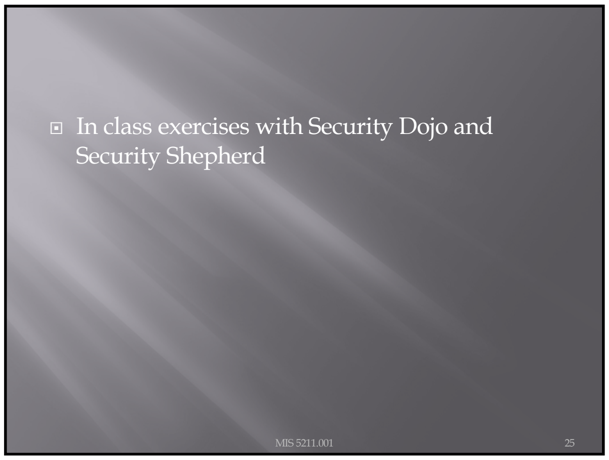
23

Famous SQL Humor

http://gizmodo.com/5498412/sql-injection-license-plate-hopes-to-foil-euro-traffic-cameras

MIS 5211.001 24

24



- In class exercises with Security Dojo and Security Shepherd

MIS 5211.001 25

25

Questions?

MIS 5211.001 26

26