Google-Hacking MySQL and More MySQL Security

Sheeri Cabral

Senior DB Admin/Architect, Mozilla

@sheeri www.sheeri.com

What is White-Hat Google Hacking?

Hacking

Using Google

White-hat

Where to Start

Do some searching

http://johnny.ihackstuff.com/ghdb

Security Advisories

App and Web servers

Applications

Companies

Google's TOS

Under 18?

No automation

What's not in the TOS

https://www.google.com/accounts/TOS

- past versions

Password Hashes

Hash Dictionaries like http://hashash.in/

Password hash is
*13824B0ECE00B527531D2C716AD36C23AC11A30B

What is the password in plaintext?

How to Use Google

wildcards * .

Different media types

Boolean search

Google Basics

10 word limit

AND assumed

foo | bar

Operators

http://www.google.com/help/operators.html

/cheatsheet.html

Site matters

filetype: vs inurl:

Google Dork

site:www.sheeri.com inurl:?id=1..100000

Vulnerable Locations

Common paths

Open source = double-edged sword

Some To Try

inurl:config.php

inurl:php?

inurl:delete

inurl:delete.php?id=

link:private.yourcompany.com

numrange:

More To Try

site:sheeri.com filetype:php inurl:id

- Then test out injection

http://*:*@www.sheeri.com

intitle:Remote.Desktop.Web.Connection site:sheeri.com

Further study

http://bit.ly/ghacks0

http://bit.ly/ghacks1

www.securityvulns.com

Defensive Strategies

Validate/scrub input

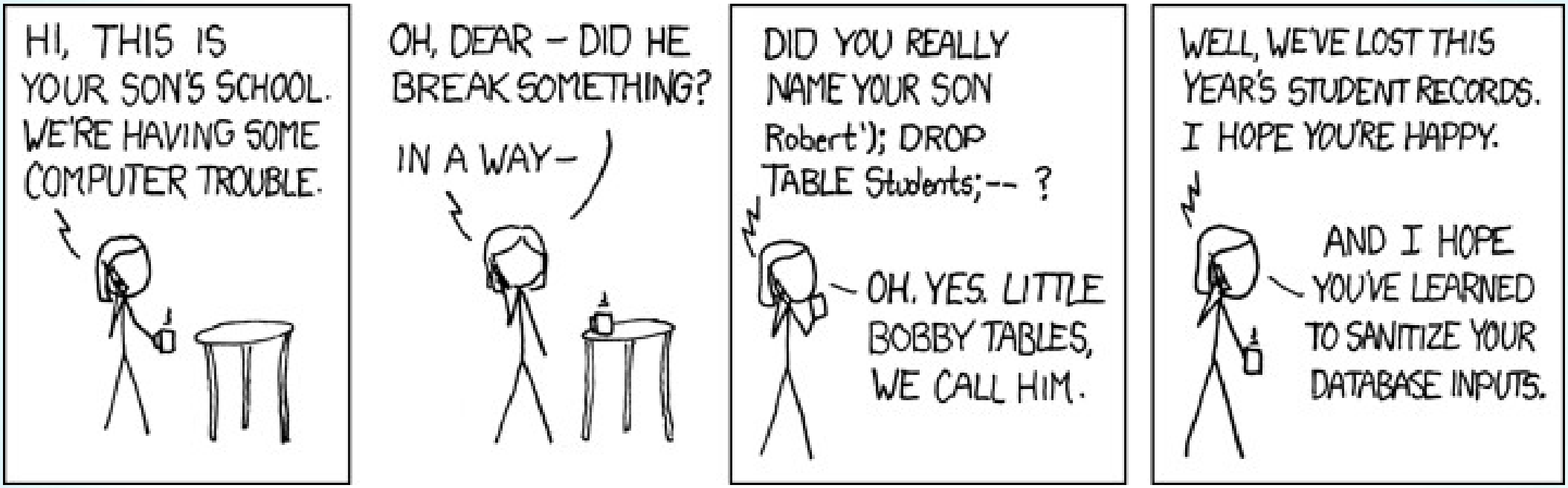
CSRF – Validate source

XSS

SQL Injection Cheat Sheet

- http://bit.ly/sqlinjcheat

SQL Injection



SQL Injection

- http://bit.ly/explainsqlinj

```
SELECT count(*) FROM users WHERE
  username='$user' and pass='$pass';

-- if count(*)>0, log in!
```

SQL Injection

- http://bit.ly/explainsqlinj

```
SELECT count(*) FROM users WHERE
  username='$user' and pass='$pass';
-- if count(*)>0, log in!
```

- Pass: hi' or 1=1

```
SELECT count(*) FROM users WHERE
  username='foo' and pass='hi' or 1=1';
```

Validate User Input

- Look for ; \g \G ' “ UNION
- HTML encoding
- NULL or char(0)
- VARCHAR and ' '

Validate User Input

- Save yourself time
- Buffer overflows
- CHARSET

Trusting GET or POST

- Only from certain pages
- cookies – even with valid session ids
- register_globals=off in PHP

When, Not If

How is application DB access stored?

As strong as your weakest link

No vaccine

Regression Testing Tools

http://sites.google.com/site/murfie/

- goolink
- crapscan
- goohosts

More Actions

Google Hacking Software

- http://code.google.com/p/googlehacks/

Google Hacks Honey Pot

- http://ghh.sourceforge.net/

Google honors robots.txt

Vulnerability Checking Tools

Goolag.org – GUI – old, but open source

Wikto/Nikto

Time for a break!

General Security

- Patching
- Prevent access
- Prevent meaningful info gathering

Access

- Network access
- Direct db access
- Access to backups

Access Points

- Who can login?
  - Network, seeing traffic
    - http://forge.mysql.com/snippets/view.php?id=15

```
shell> tcpdump -l -i eth0 -w -src or
  dst port 3306 | strings
```

  - OS
    - Data
    - Logs
    - Backups

Operating System

- Authentication
- Firewall
- Other installed programs

Securing your Application

- Authentication
- Config files
- User-entered data
  - SQL injection

Who has access?

- pt-show-grants
- SELECT user, host, length(password), ssl_type FROM mysql.user

  WHERE Super_priv='Y'

  WHERE user=''

Where is the access from?

- %
- %.company.com
- 10.0.% or 192.168.%
- 10.0.1.% or 10.0.1.10_

GRANTing Access

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ...
    ON [object_type]
        {tbl_name | * | *.* | db_name.* | db_name.routine_name}
    TO user [IDENTIFIED BY [PASSWORD] 'password']
    [REQUIRE        NONE |      [{SSL| X509}]
   [CIPHER 'cipher' [AND]]       [ISSUER 'issuer' [AND]]
        [SUBJECT 'subject']]   [WITH with_option [with_option] …]
```

http://dev.mysql.com/doc/refman/5.5/en/grant.html

ACLs – to do what?

- --local-infile=0
- --skip-symbolic-links
- GRANT
  - MAX_QUERIES_PER_HOUR
  - MAX_UPDATES_PER_HOUR
  - MAX_CONNECTIONS_PER_HOUR

Other ACL's

- Object access
- Password policies
- Roles

Access from...?

- localhost only, --skip-networking
- firewall
- Who can [attempt to] DOS you?

Changing ACLs

- Who changes ACLs?
- How are ACL changes audited?
- When do ACL changes happen?

Securich

- Darren Cassar, http://www.securich.com/
- Create/drop roles

```
call create_update_role('add','role1','select');
```

Create users with roles, adding objects

Drop users, revoke privileges

```
call grant_privileges('username','hostname', 'databasename',
'tablename','tabletype','rolename', 'email');
```

```
call grant_privileges('john','machine.domain.com',
 'employees', '','alltables','role1', 'john@domain.com');
```

Securich

- Reserved usernames
- Block users
- Rename users
- Clone users
- Reconciliation

Server Options

- --bind-address
- --skip-name-resolve
- --skip-show-database

Test Database

- Anyone can access it
- Stuff with data

OS Files and Permissions

- mysql server user
- mysql server files & logs
- Passwords on commandline
- Office policies/runbook

How Does Your Data Flow?

- Where is data encrypted?
- Where do errors go?
  - Are those logs checked?
- Where does the traffic flow?

Separating Admin Apps

- Same data, different interface
- Performance, e.g. reporting
- Only allowed from VPN?
  - Public vs. easily accessible

Plaintext information

- Passwords
- Credit card info
- Identification numbers (e.g. SSN)

Hashes

- Passwords

*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 = 'password'

- Be careful where encrypted traffic goes!

Stored Code

- Stored procedures / functions
- Views
- Events
  - Instead of cron

Prepared Statements

```
PREPARE stmt1 FROM 'SELECT uname FROM
    UserAuth WHERE uname=? and pass=?';

SET @a = "alef";  SET @b = md5("alef");

EXECUTE stmt1 USING @a, @b;
```

Prepared Statements

```
PREPARE stmt1 FROM 'SELECT uname FROM
    UserAuth WHERE uname=? and pass=?';
SET @a = "alef";  SET @b = md5("alef");
EXECUTE stmt1 USING @a, @b;
```

```
SET @a = "alef";
SET @b = "alef' or 'x'='x";
EXECUTE stmt1 USING @a, @b;
DEALLOCATE PREPARE stmt1
```

Prepared Statements Perl Code

```
$query = $sql->prepare("SELECT uname FROM
UserAuth WHERE uname = ? AND pass = ?");
$query->execute($uname, $pass);
```

Prepared Statements PHP Code

```
$stmt = $mysqli->prepare("SELECT uname
  FROM
   UserAuth WHERE uname = ? AND pass = ?");
$stmt->bind_param($uname, $pass);
$stmt->execute();
```

Prepared Statements Java Code

```
PreparedStatement pstmt =
  con.prepareStatement("SELECT uname
  FROM
  UserAuth WHERE uname = ? AND pass
= ?");
  pstmt.setString(uname, pass);
  ResultSet rset = pstmt.executeQuery();
```

Prepared Statements .NET/C# Code

```
using(SqlCommand cmd = new SqlCommand("SELECT
   uname FROM UserAuth WHERE uname = @uname
   AND pass = @upass",con)) {

        cmd.Parameters.AddWithValue("@userName",
          userName);

        cmd.Parameters.AddWithValue("@pass", pass);

            using( SqlDataReader rdr =
          cmd.ExecuteReader() ){

            ...}

          }
```

Auditing and Monitoring

- Prevention is one part of security
- Auditing - review and assess security
- Monitoring – alerting of security issues

Auditing and Monitoring

- General log to see all login attempts
- Locking out accounts with max_connect_errrors
  - global

Play hard to get

- MySQL Events instead of cron/task scheduler
- NO PLAINTEXT PASSWORDS
- Do not store it if you do not need it

Authentication Plugin

- MySQL 5.5 (since Dec 2010)
- MySQL Enterprise Plugins
  - Windows Authentication
  - PAM Authentication

Creating Policies

- There will be exceptions
  - But it's still a good idea to have the policies!

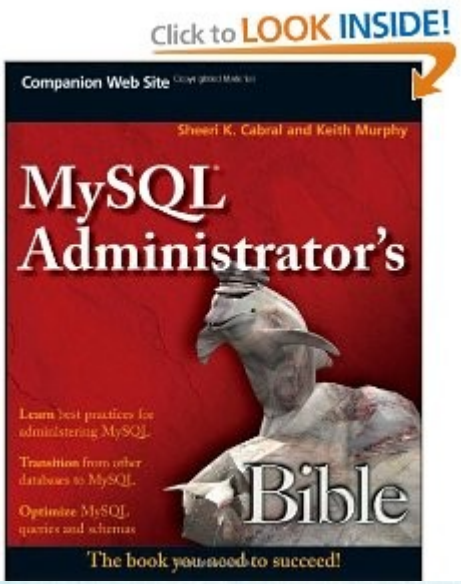
Questions? Comments?

OurSQL podcast

- www.oursql.com

MySQL Administrator's Bible

- tinyurl.com/mysqlbible

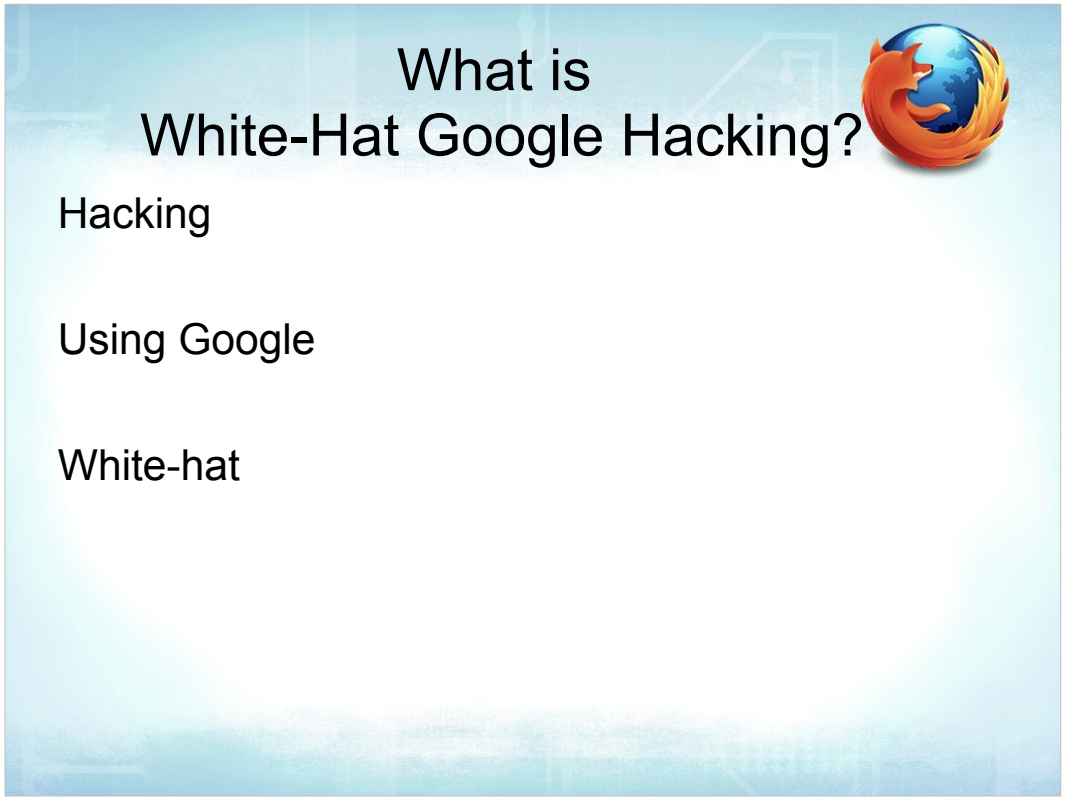


kimtag.com/mysql

planet.mysql.com

Google-Hacking MySQL and More MySQL Security

Sheeri Cabral
Senior DB Admin/Architect, Mozilla
@sheeri www.sheeri.com



By “hacking” I mean poking around to see if your site has security vulnerabilities.

Google hacking uses Google to research. For instance, if your site runs “wordpress”, you can search for:

wordpress security vulnerability exploit

site:sheeri.com wordpress

White hat – meaning the good, legal kind.

Because Google caches pages too, you can find information. This also means that other archive sites can be useful. You may take “powered by wordpress off your site, but once you know about it....!” not images though.

Sometimes it helps to see what's already out there.

info:www.sheeri.com

Go to:
http://johnny.ihackstuff.com/ghdb/

click on “error messages”

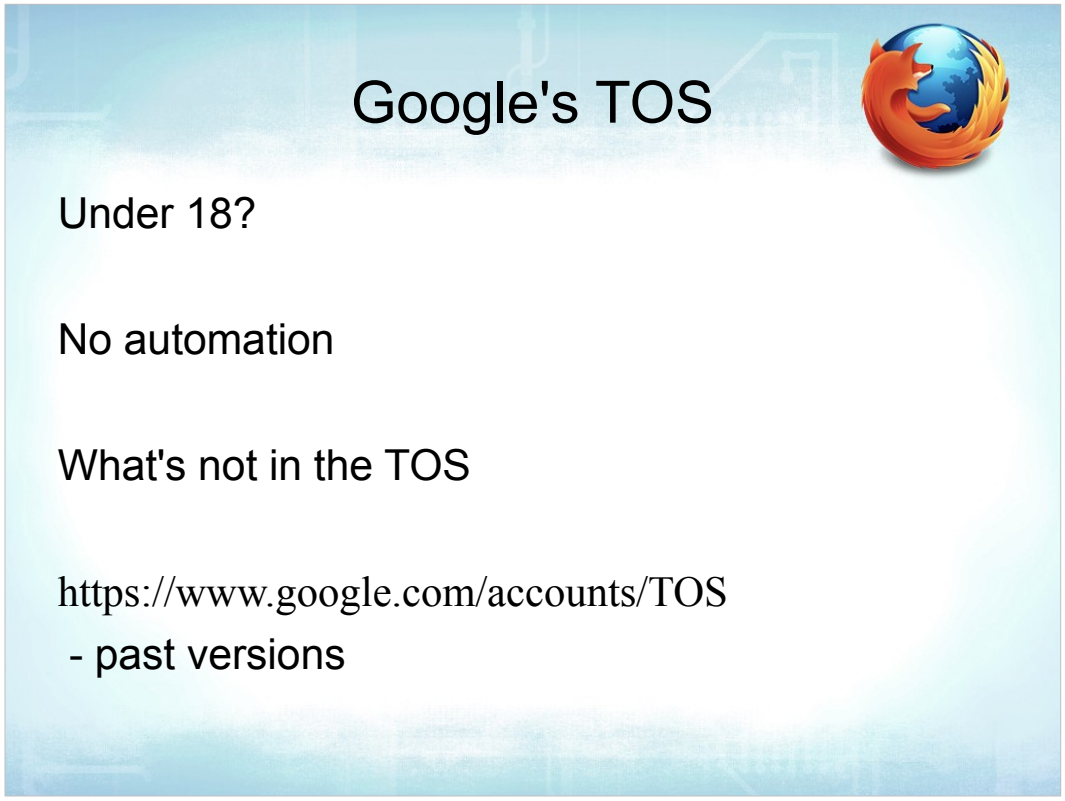
Show a few

for the impatient, search Google:
site:johnny.ihackstuff.com mysql

Note that you'll be searching your site only, but hackers will be searching for specific vulnerabilities.

site:sheeri.com "powered by wordpress"

site:www.sheeri.com MySQL.Error



https://www.google.com/accounts/TOS

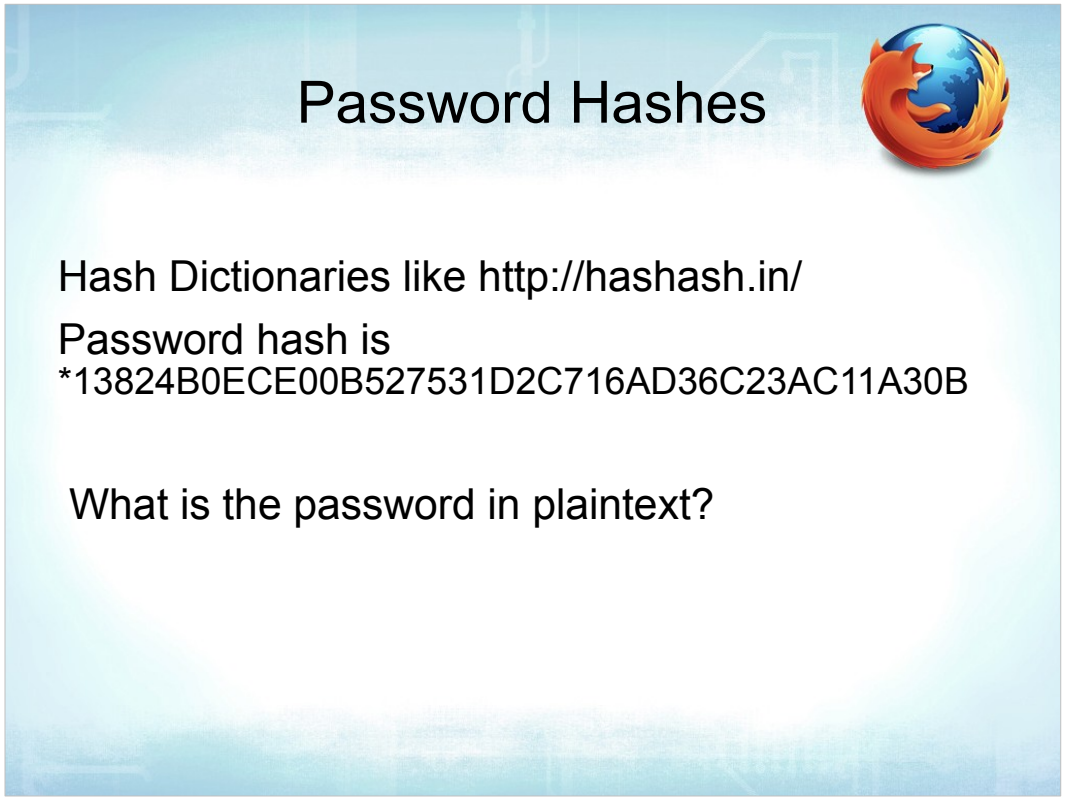
If you're under 18, please don't use Google. (although they have the magic clause 20.5 – if one part is bad the rest of the contract is still good)

Section 4.5 – number of transmissions or data storage – so if you're automating searches and retrievals, you want to throttle yourself

Section 5.3 -- don't even try to automate!

What's not in the TOS -- “don't break laws using Google's services”. That being said....don't!

13.3 (B) Google is required to do so by law (for example, where the provision of the Services to you is, or becomes, unlawful);



https://www.google.com/accounts/TOS

If you're under 18, please don't use Google. (although they have the magic clause 20.5 – if one part is bad the rest of the contract is still good)

Section 4.5 – number of transmissions or data storage – so if you're automating searches and retrievals, you want to throttle yourself

Section 5.3 -- don't even try to automate!

What's not in the TOS -- “don't break laws using Google's services”. That being said....don't!

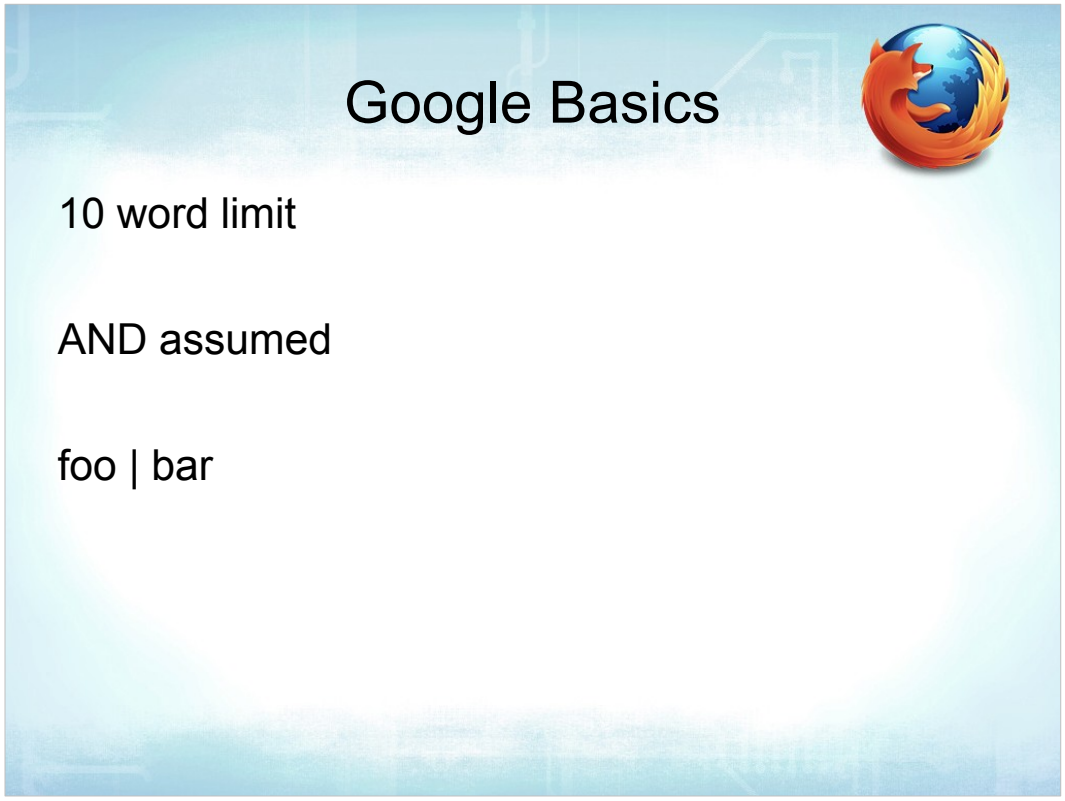
13.3 (B) Google is required to do so by law (for example, where the provision of the Services to you is, or becomes, unlawful);

* is 1 word missing, . Is 1 character.

Not bad, because Google automatically does the stemming you want (ie, database vs databases)

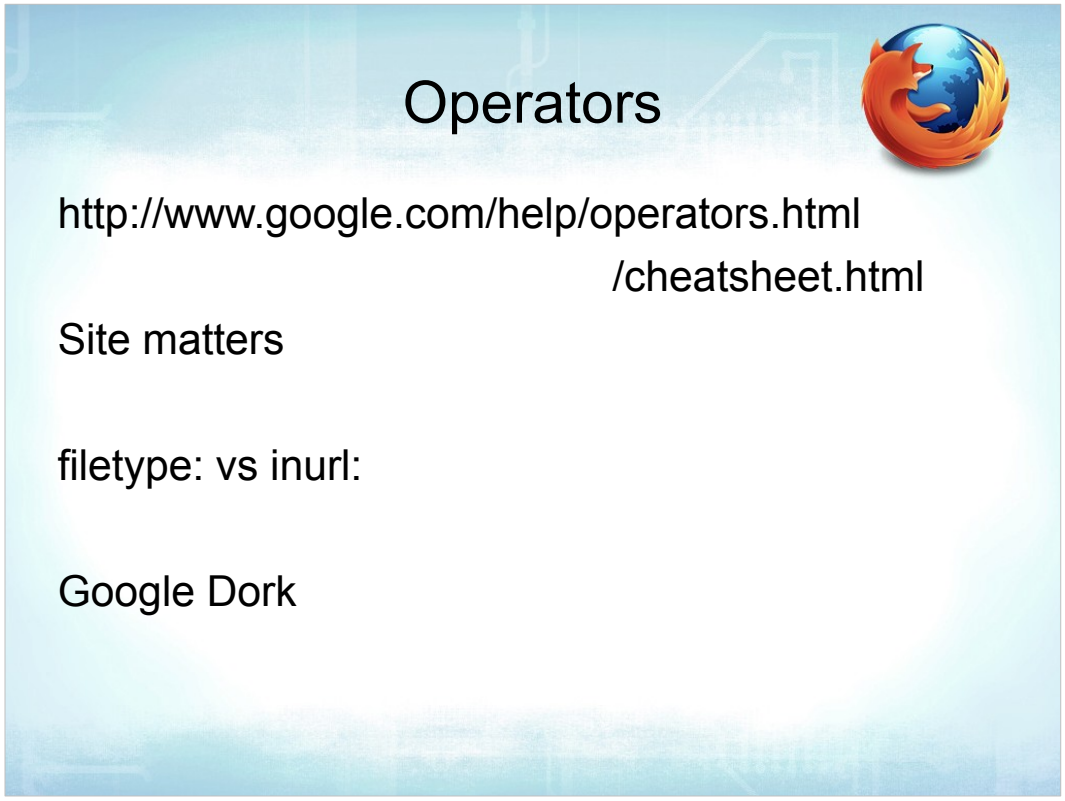
different media types – blog search vs. news search, etc.

+ will force a result, if you want a common word like “a” - will force no result with that. Also, quotes around things do to exact matches.



Foo bar searches for “foo” and “Bar”

but foo | bar searches for either or. UNION of results.
Similar likely because Google weighs relevance....



http://www.google.com/help/operators.html

we already mentioned site:

site:sheeri.com viagra

site:www.sheeri.com viagra

site:sheeri.net viagra (same)

site: sheeri.org viagra

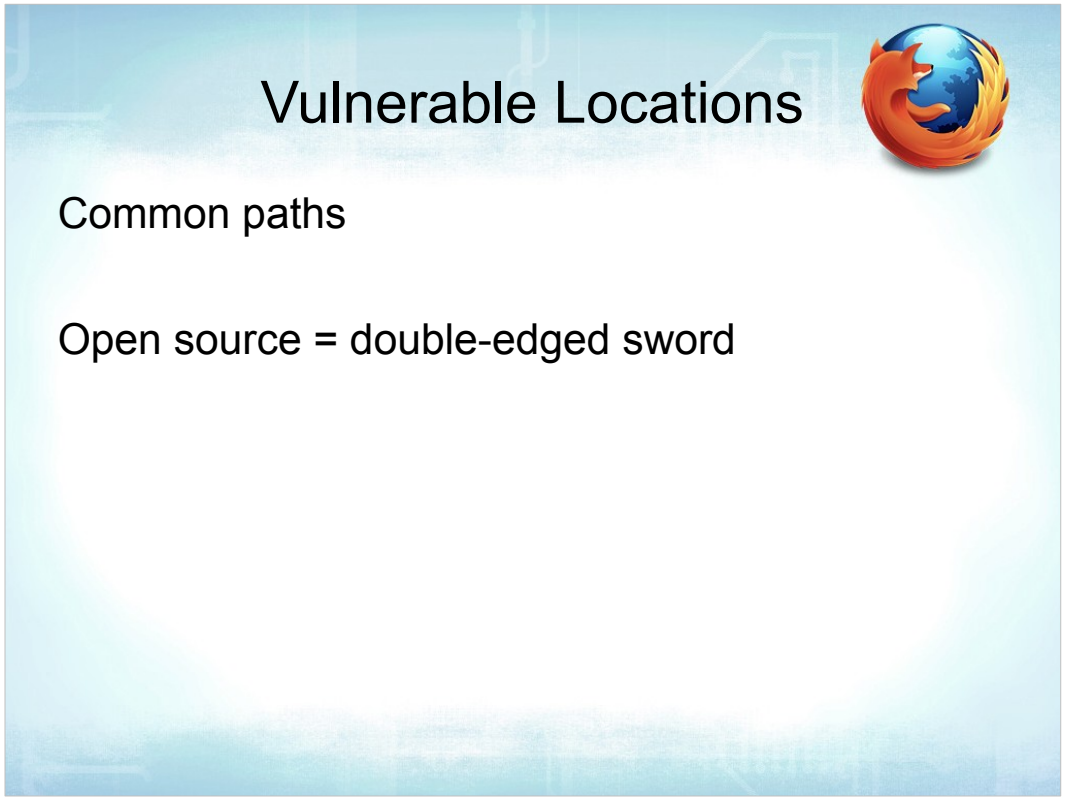
So try out all your domains -- I can't use “inurl:sheeri”

inurl:sheeri viagra

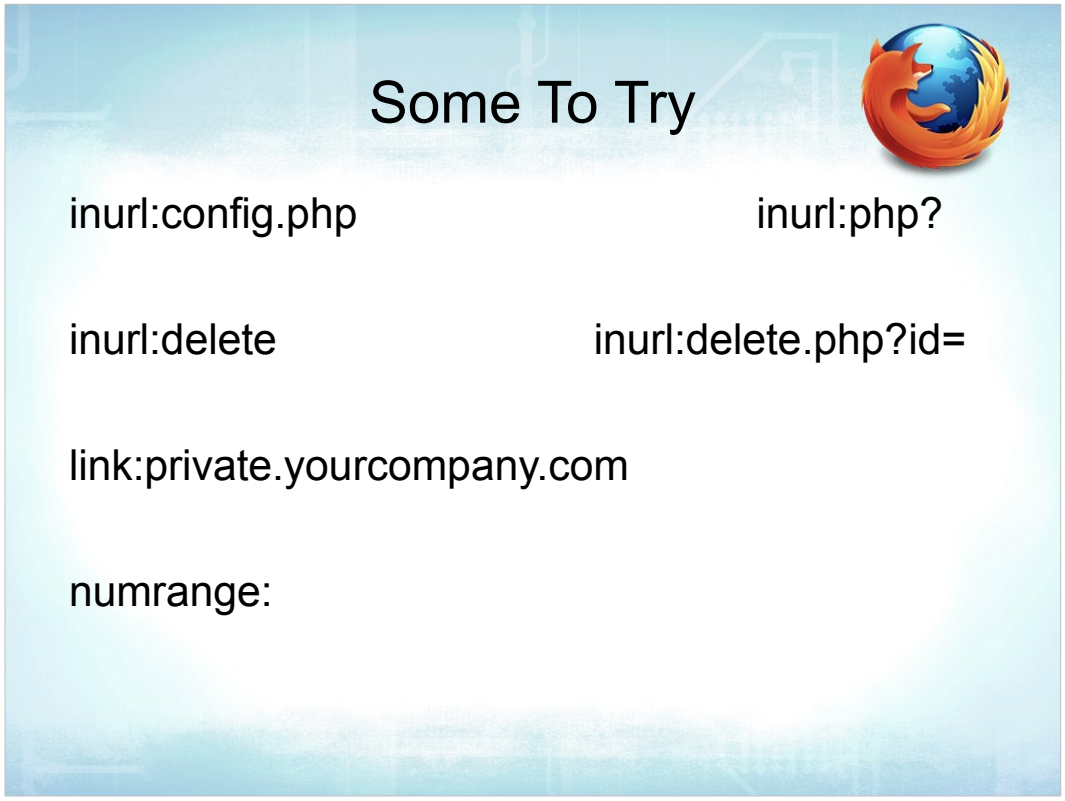
You can do “Filetype:” for php files, html, jsp, etc but can also use “inurl”

intitle:index.of site:www.sheeri.com for dir listings


site:www.sheeri.com inurl:?id=1..100000



site:sheeri.com inurl:admin



site:www.sheeri.com inurl:config.php
site:www.sheeri.com inurl:admin.php

site:www.sheeri.com inurl:"php?"

shows variables

inurl:delete – if you're sending the actions with a GET variable, that's bad!
There's also delete.php

is there a site that is linking where it shouldn't?

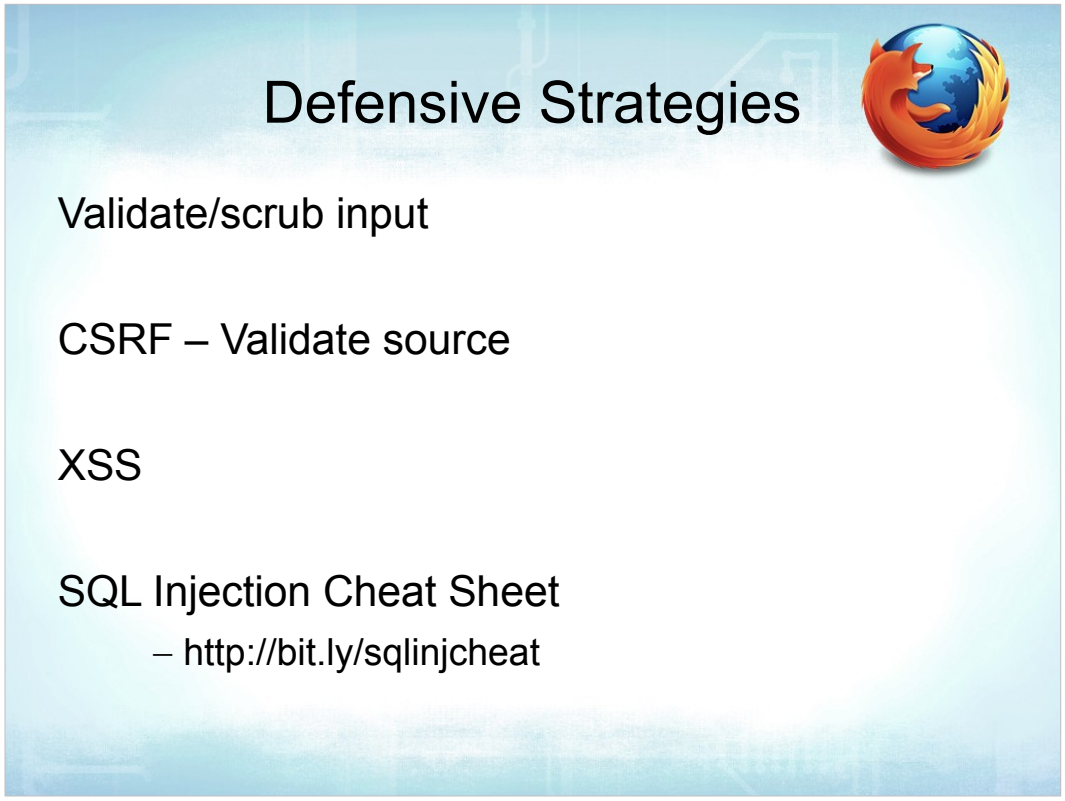
credit card – number ranges

5) "site:<your site> filetype:php inurl:id" - By searching for files of type php, you can sometimes find applications that are accepting parameters by looking for "id" in the URL. Then, use a trick I got from Erratasec, replace the fields with ' and find many SQL injection vulnerabilities.

: is for user:pass

5) "site:<your site> filetype:php inurl:id" - By searching for files of type php, you can sometimes find applications that are accepting parameters by looking for "id" in the URL. Then, use a trick I got from Erratasec, replace the fields with ' and find many SQL injection vulnerabilities.

: is for user:pass



Only use what's needed, to avoid query injection, and use prepared statements when possible, you can also now use them in conjunction with stored procedures so the query is handled by the db code, instead of having the developers write code.

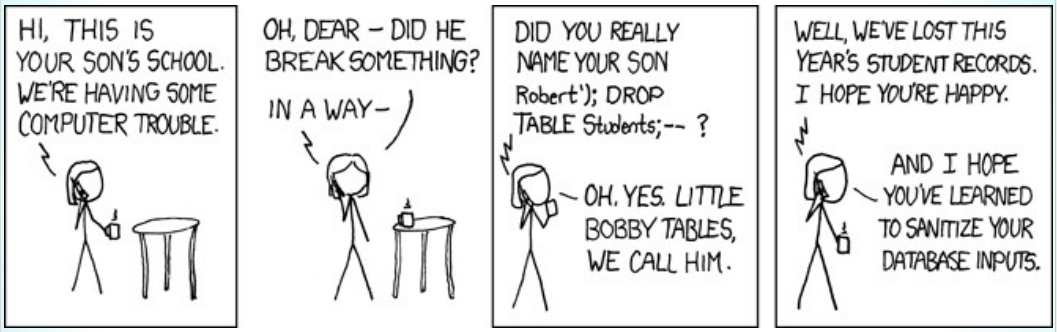
This doesn't help when someone goes through and pulls up account information for customer 1, customer 2, etc (or deletes them). That is CSRF – Cross Site Request Forgery -- uses completely valid requests.

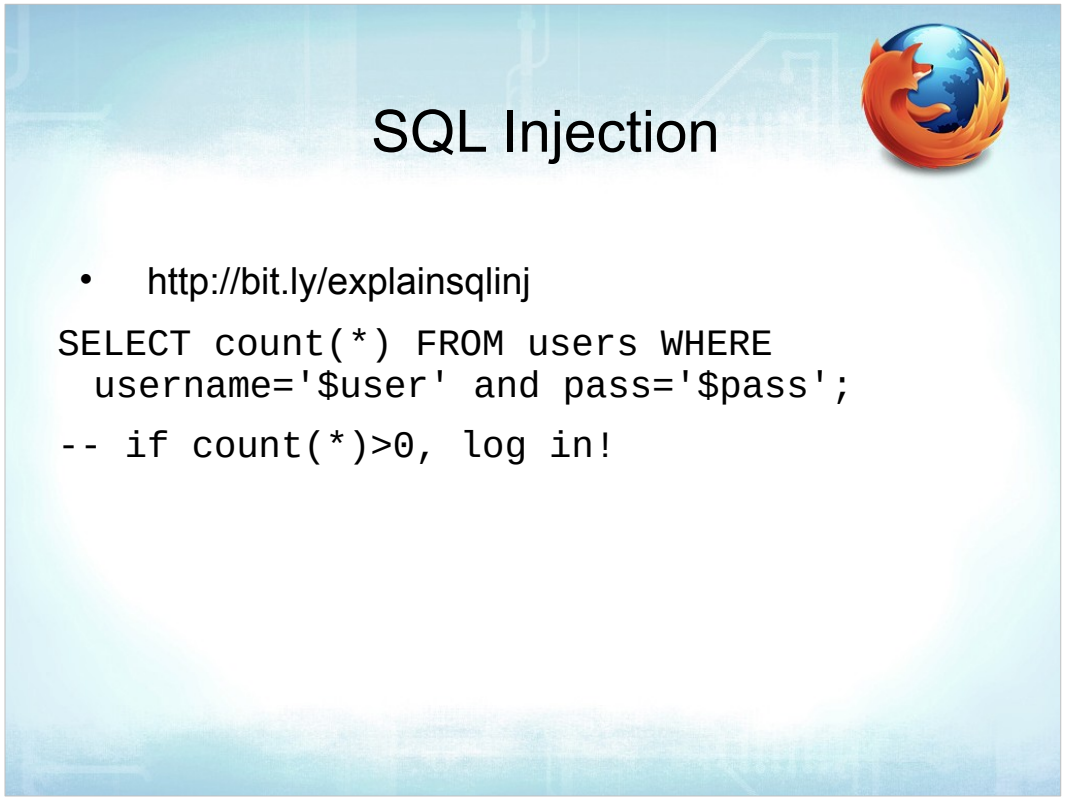
To defend against that, referer checking (hackable) or validation tokens (for site and for permission—do not think "if they got to this page they can execute the code"—re-validate if necessary)

XSS = cross-site scripting, ie using a form for SQL injection.

SQL Injection Cheatsheet: http://ferruh.mavituna.com/makale/sql-injection-cheatsheet/

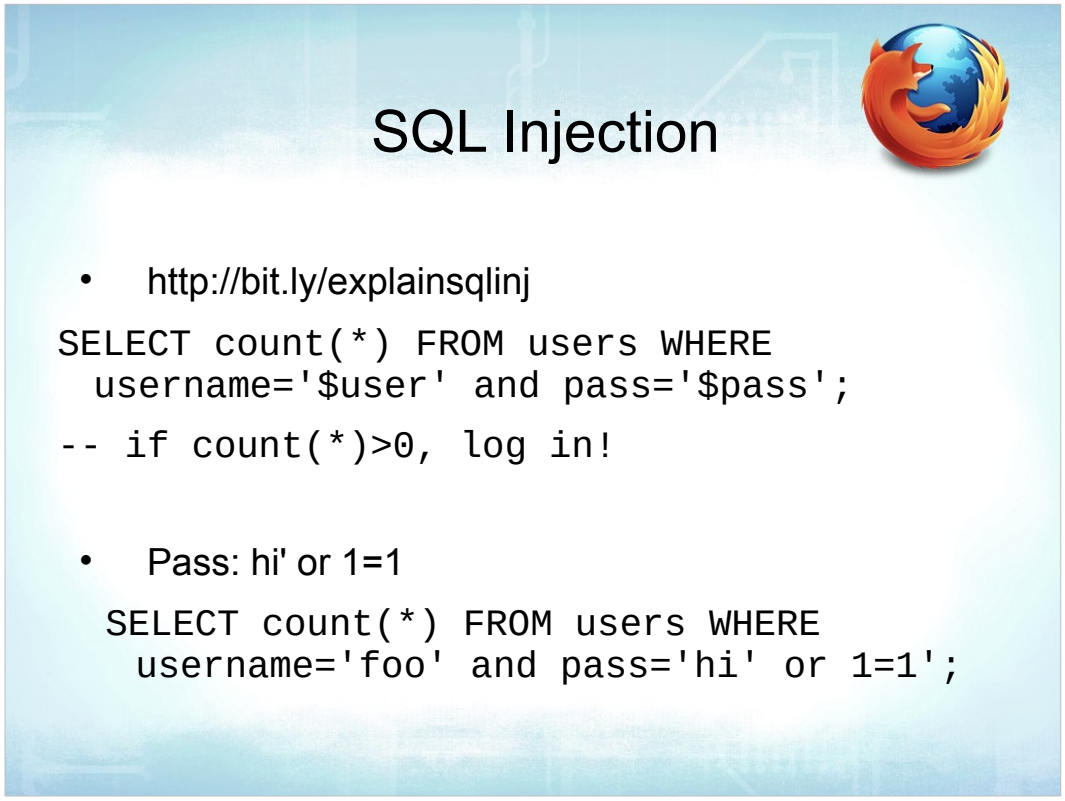
SQL Injection





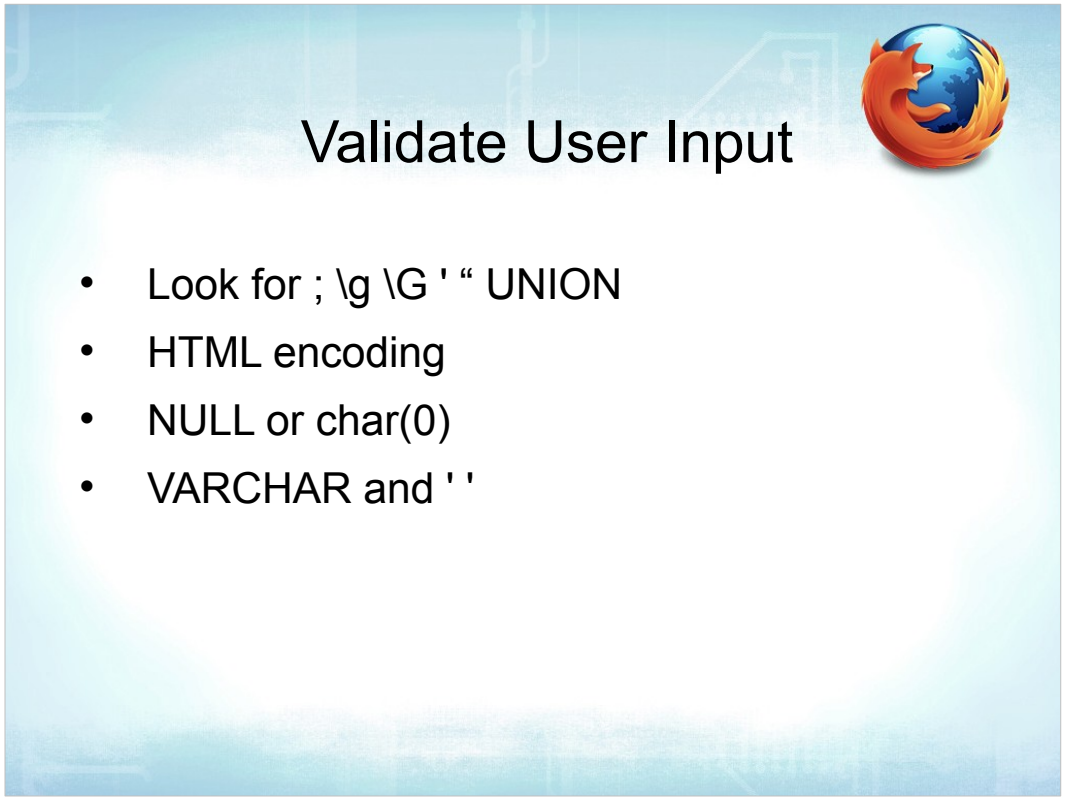
I'm not going to talk much about SQL injection, but I'll give an overview:

Let's say you put in your password



I'm not going to talk much about SQL injection, but I'll give an overview:

Let's say you put in your password



Disallow or escape ; \g \G “ ' UNION (; won't always help, check if multi_query is allowed)

XSS - Do you allow HTML in stored forms? Including javascript? Personal ad and <G> in form renders weird. Not to mention <SCRIPT .... folks put links to their pay-per-click ads, whenever their page is clicked...

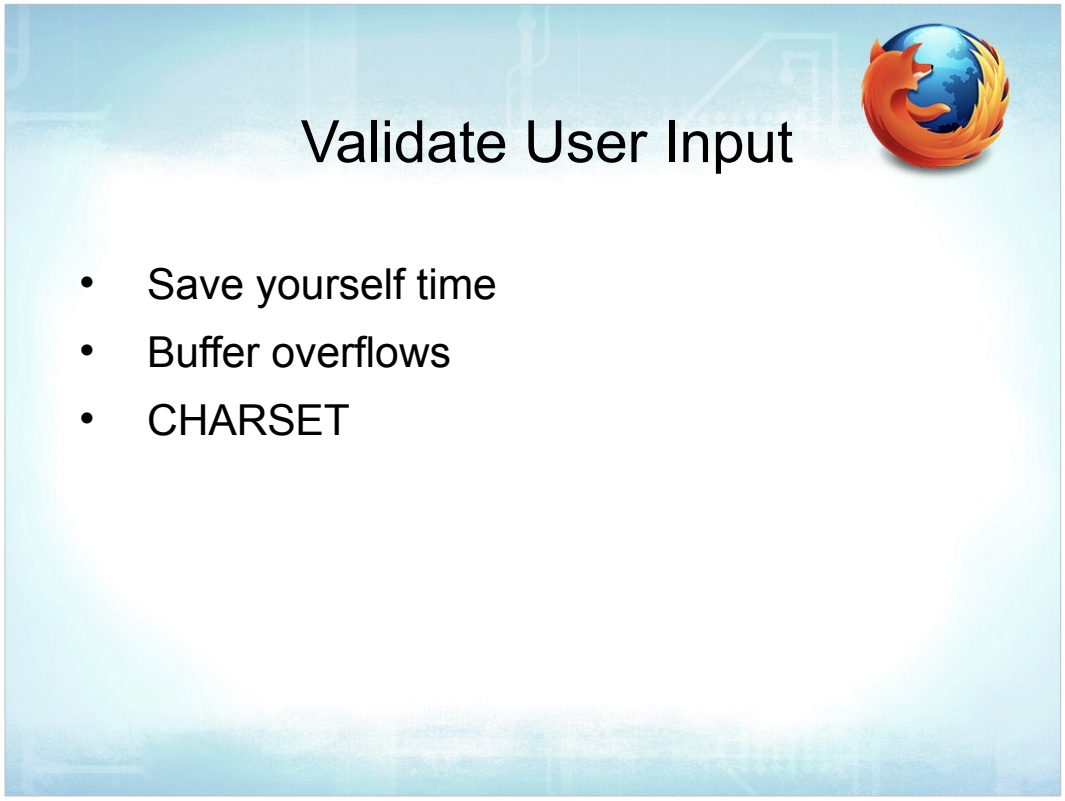
Type 0 XSS -- ?? page's client-side script, ie javascript, access URL request and uses info on that page for something in the current page, can be exploited – can put in another script.

Type 1 XSS – server gets data from client, client can put scripts in there. Reason to strip out HTML

Type 2 XSS – when this stuff is stored.

NULL / char(0) (mysql_query("/*".chr(0)."*/ SELECT * FROM table");)

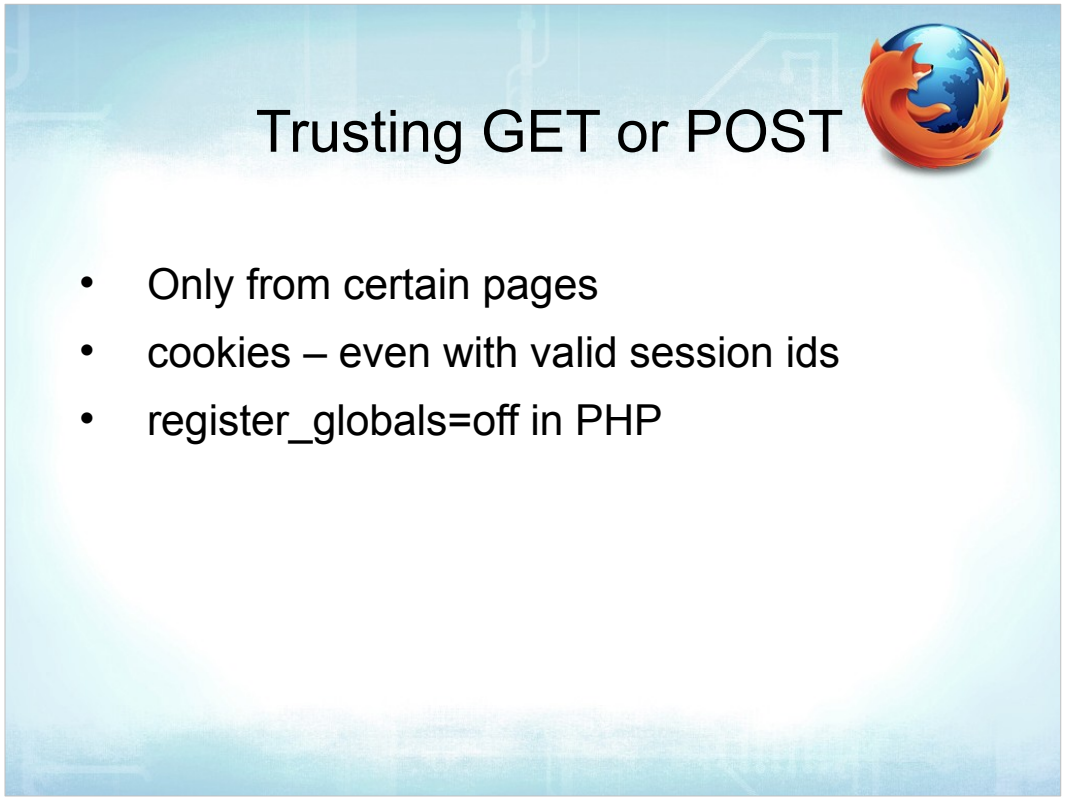
' ' and varchar



Save yourself time, include e-mail checks if you can (php checkdnsrr)

Buffer overflows

What's your CHARSET? (length of INPUT TYPE=TEXT != # of bytes!)



Easy to copy your web form and send it

HIDDEN fields too all you have to do is view source!

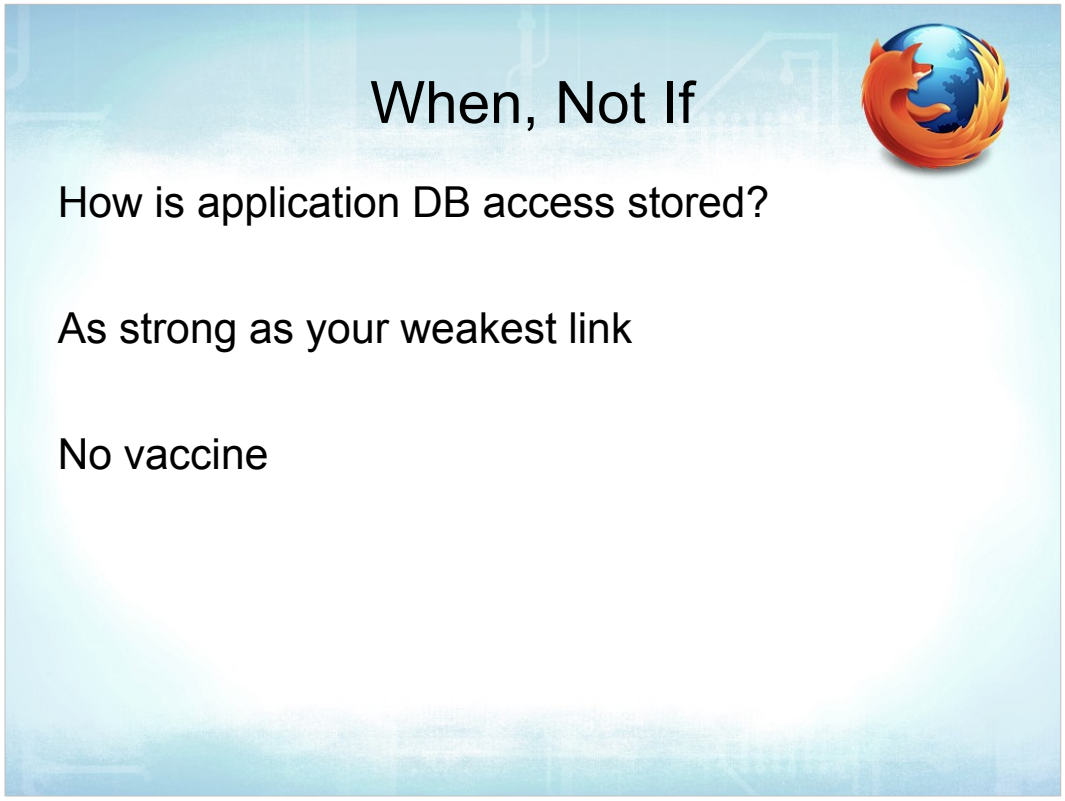
Valid user can do bad stuff, so even with a session ID don't trust unless it's your site

register_globals off in php to avoid POST params in GET context

index.php?$auth=true

Buffer overflows

What's your CHARSET?



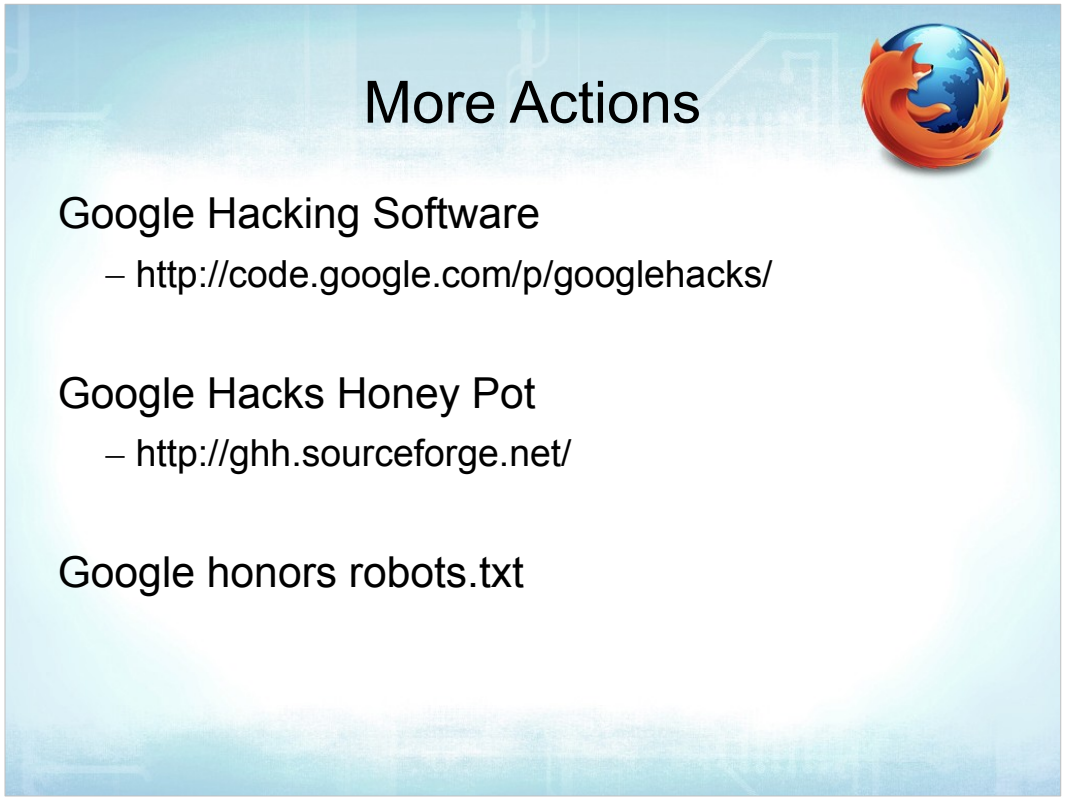
And that weak link might be someone putting passwords on an intranet wiki they didn't realize was being searched by google!

There is no vaccine – if you're using old software, you have to upgrade. Just like viruses and worms resurface, because of the nature of the web it's not like people are going to “forget” vulnerabilities.

Goolink -- parse all the hyperlinks in a saved google search results page so they can be downloaded with 1 command (wget -i results.html) or they can be used with other scripts (hostlookup etc..)

Crapscan – searches for certain files in a URL tree. You can customize the files, like “apache_log” -- for regression testing

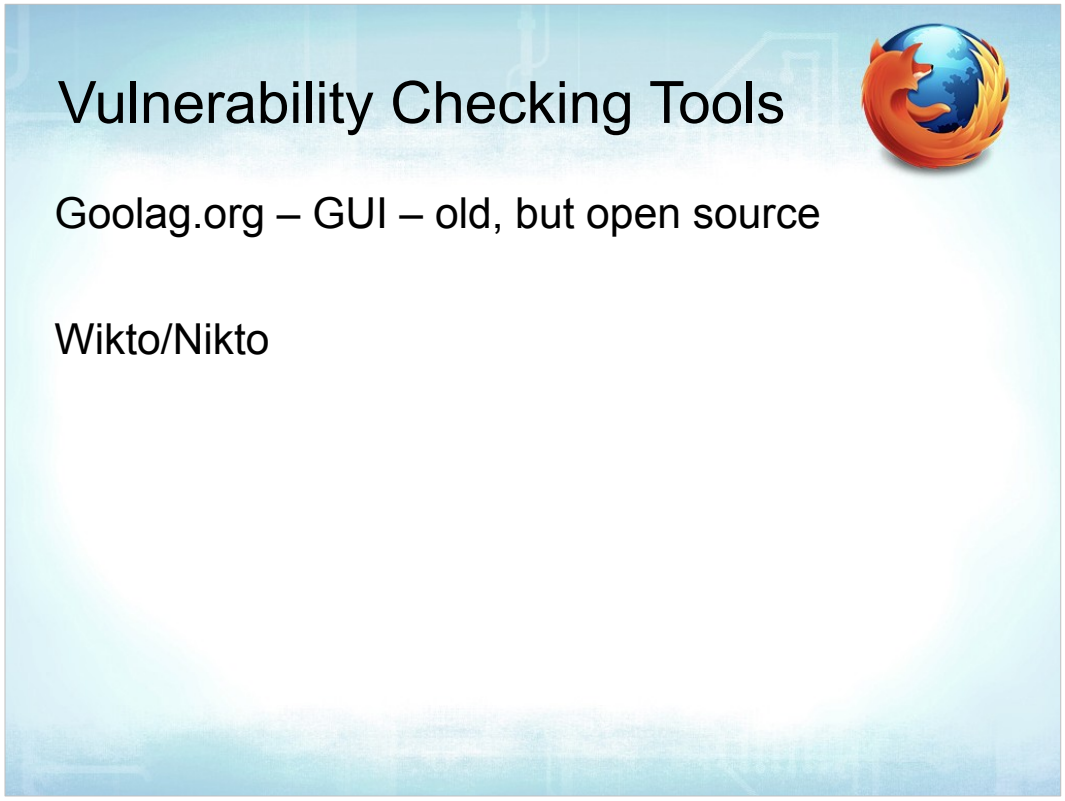
goohosts – check webserver header response – cygwin version, did a quick check and couldn't find the original, don't know if it's linux or what.



....

use the honey pot to trap people and find them if you have the time.

You can have your pages removed from Google, and Google honors the robots.txt, but most of us don't want that. www.robotstxt.org



http://www.goolag.org/specifications.html

Windows, .NET framework. GUI-based, type in a host and a list of things to check. When I installed it voices came up, so be prepared. 10 dorks or less to scan, it doesn't warn, otherwise it does.

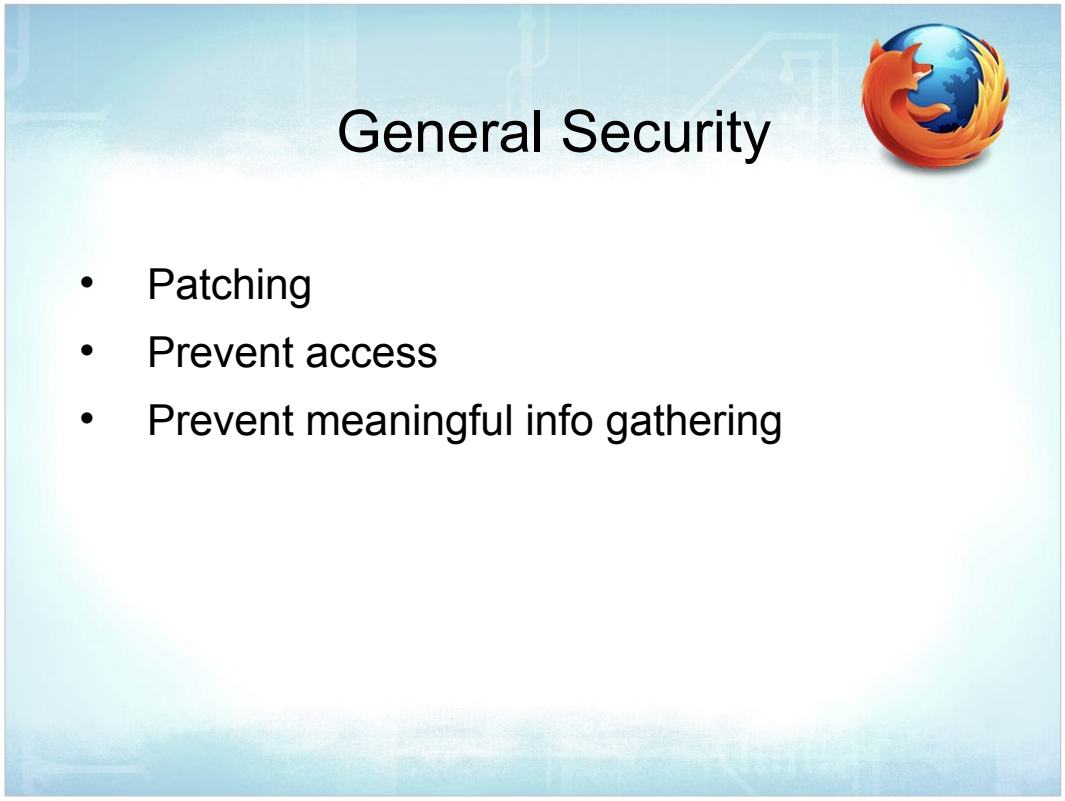
Scan for “powered by wordpress” on www.sheeri.com

Web server assessment tools.

http://www.sensepost.com/research/wikto/

http://www.cirt.net/nikto2

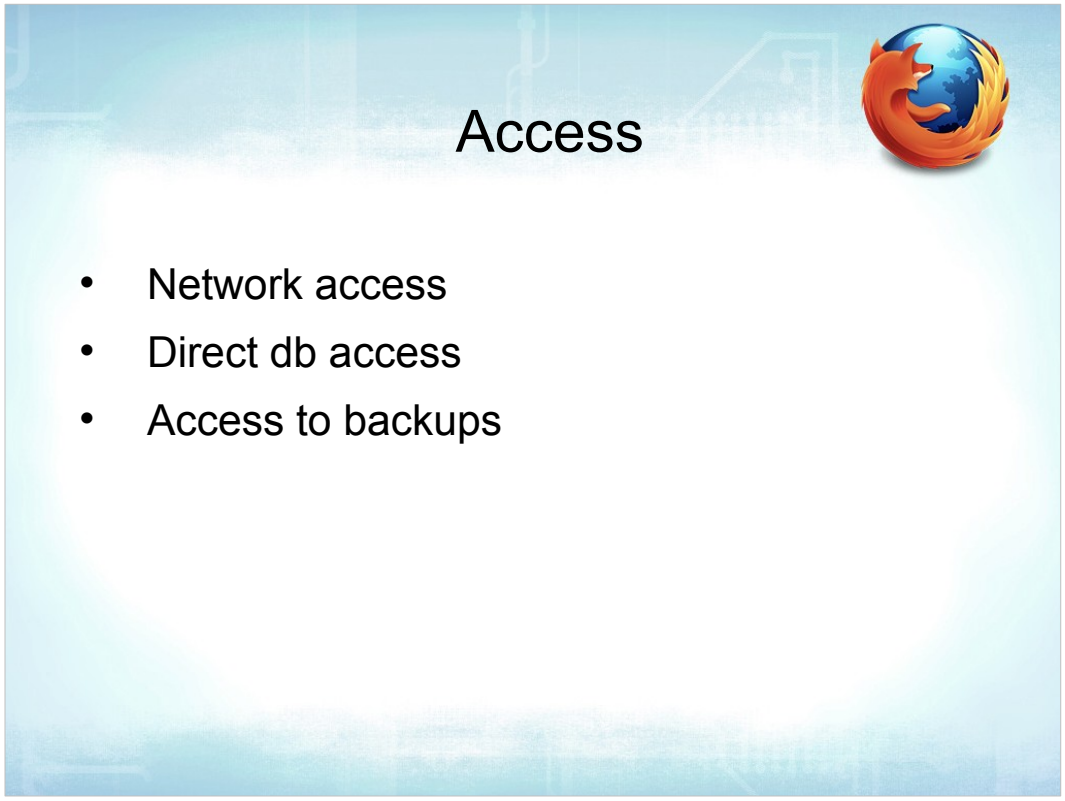
Time for a break!



MySQL has a new version each month! Can't patch every month, but you should upgrade every 6-12 months.
MySQL 5.1 GA Nov 2008
MySQL 5.5 GA Dec 2010

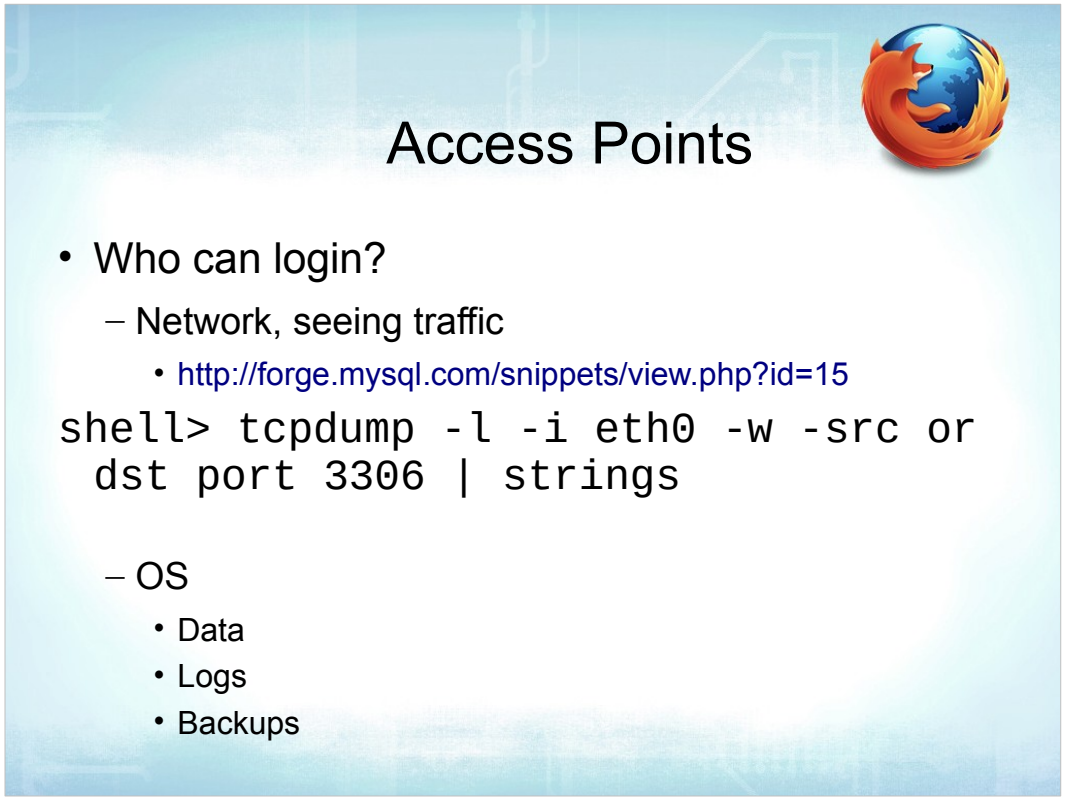
As for meaningful info gathering, e.g. encryption!

Preventing access includes permissions and ACL's but it's not limited to that



Can someone sniff traffic going across the network? What about replication or backups?

Can anyone try to login to port 3306 with telnet?

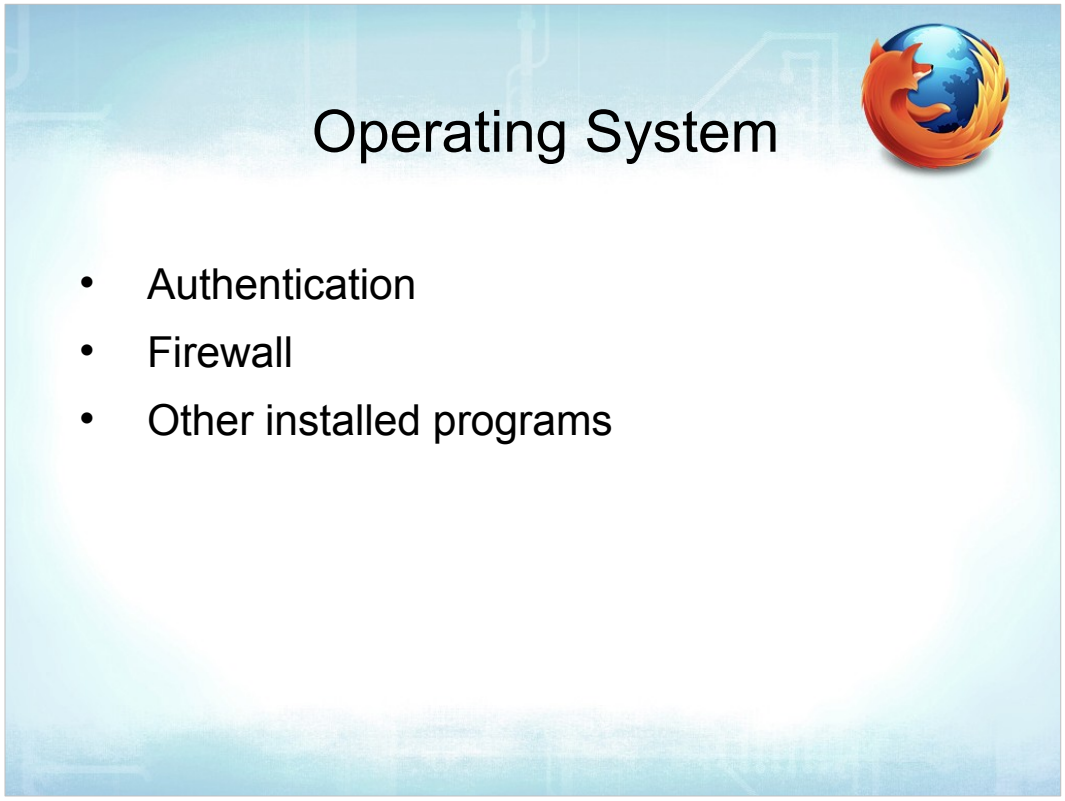


Poor man's query profiler

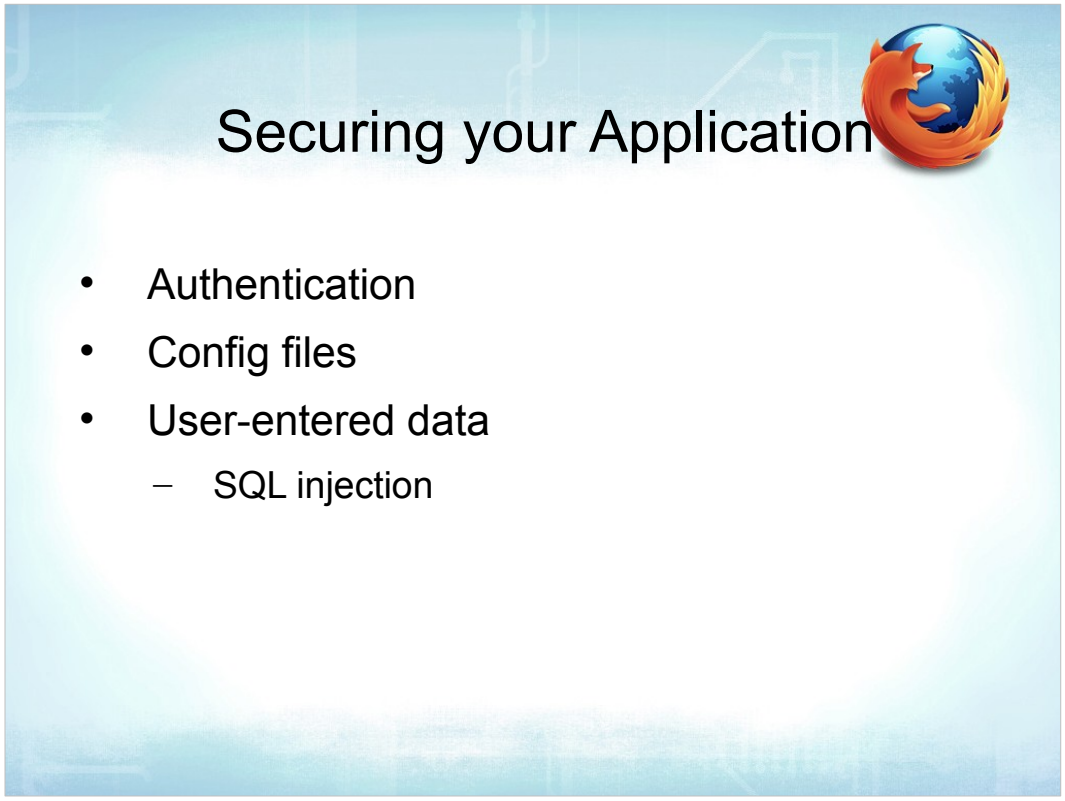
Who can login to the OS and see the data? Strings on a MyISAM table can get data!

Who can login and read the logs? Slow, binary logs?

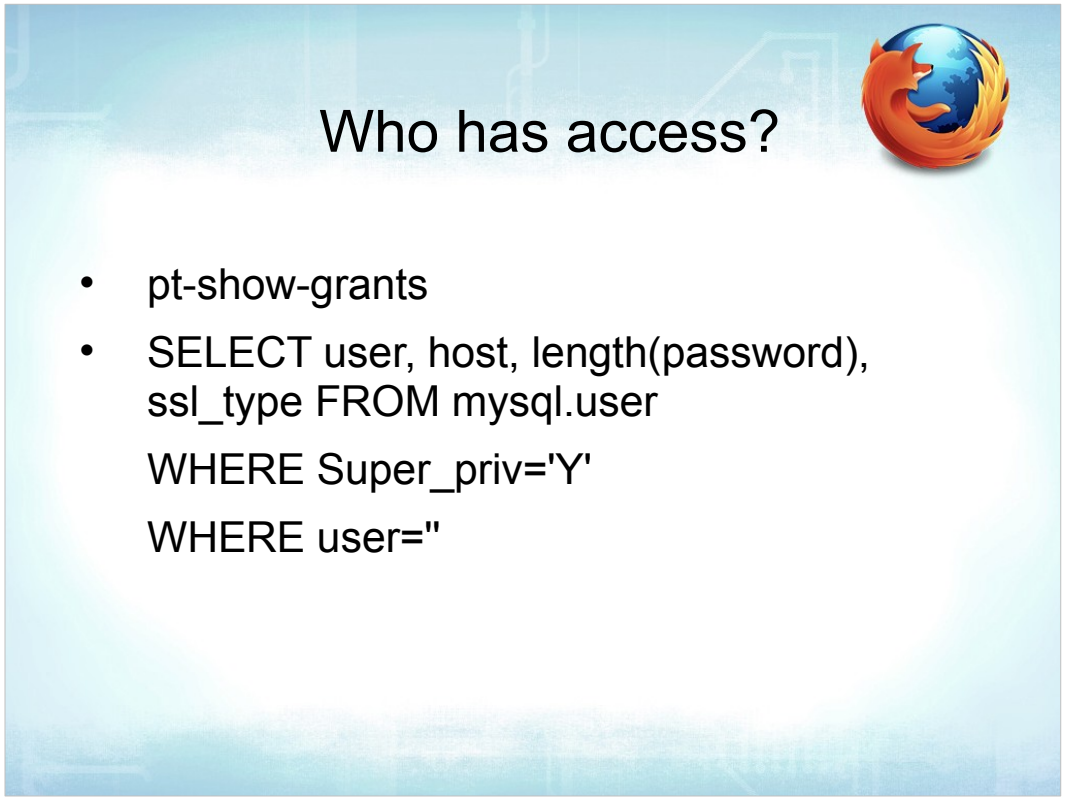
Read the backups?



Are there other installed programs that are running on the same user, such as “nobody”? What other people have access due to the other installed programs?

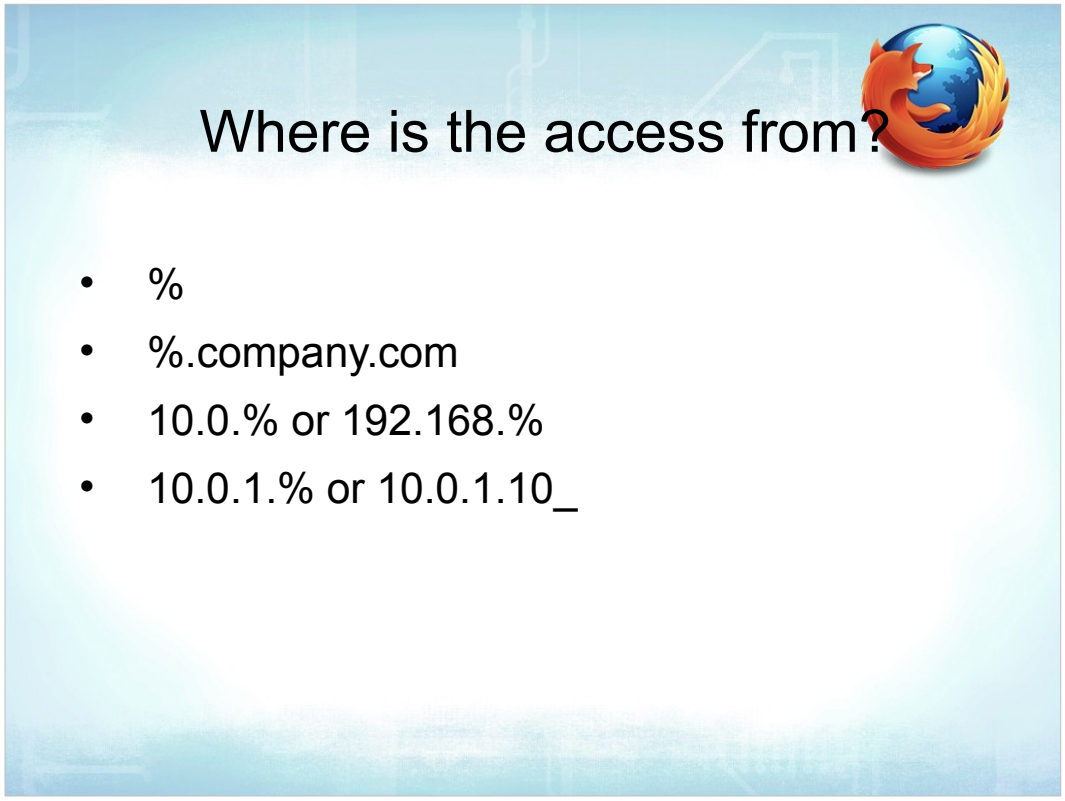


I talk about SQL injection and authentication more in-depth



Ways to see who has access

Super_priv can shutdown with mysqladmin shutdown. Also can write even if db is read_only. Also if max_connections is reached, 1 more user can login but only if they have the super priv.



This gets tricky because IP's can be spoofed, and if you're using Amazon EC2 or other cloud solutions (including traditional shared hosting) your IP might change without notice.

GRANTing Access

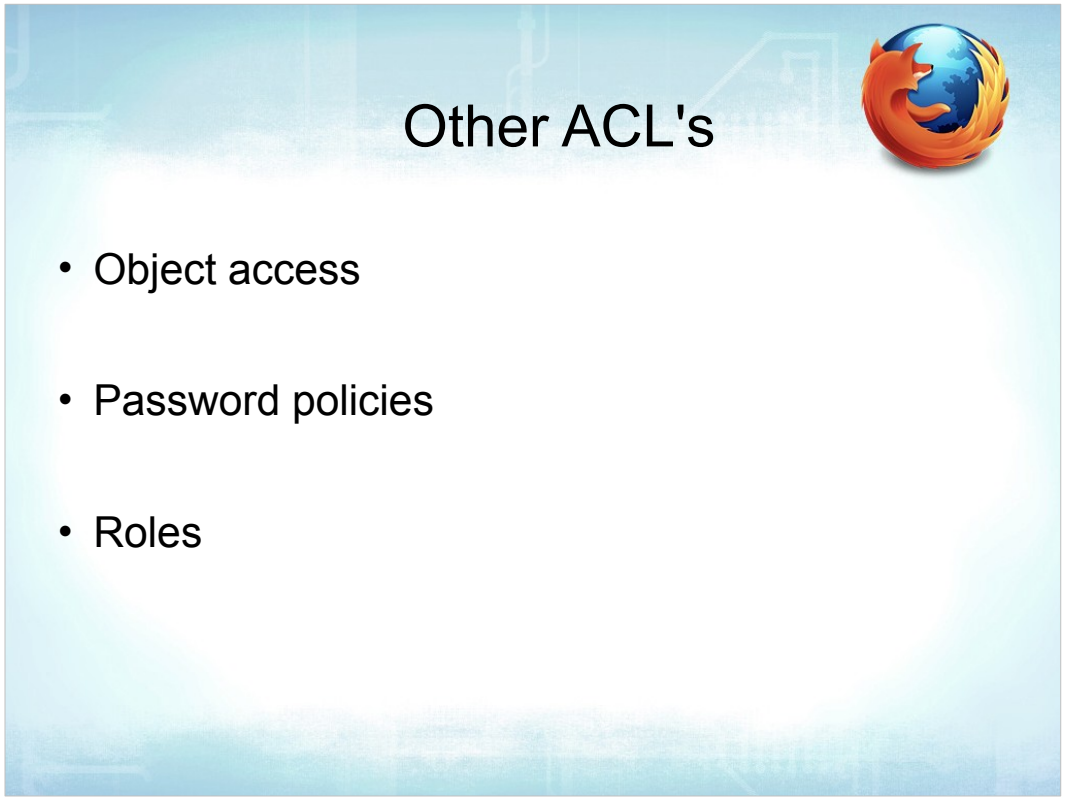
```
GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ...
    ON [object_type]
        {tbl_name | * | *.* | db_name.* | db_name.routine_name}
    TO user [IDENTIFIED BY [PASSWORD] 'password']
    [REQUIRE        NONE |      [{SSL| X509}]
   [CIPHER 'cipher' [AND]]       [ISSUER 'issuer' [AND]]
        [SUBJECT 'subject']]   [WITH with_option [with_option] …]
```

http://dev.mysql.com/doc/refman/5.5/en/grant.html

priv_type is the most important thing here, show the doc with the charts in it.

ACLs – to do what?

- --local-infile=0
- --skip-symbolic-links
- GRANT
  - MAX_QUERIES_PER_HOUR
  - MAX_UPDATES_PER_HOUR
  - MAX_CONNECTIONS_PER_HOUR



Who can acces stored procedures/functions? Views?

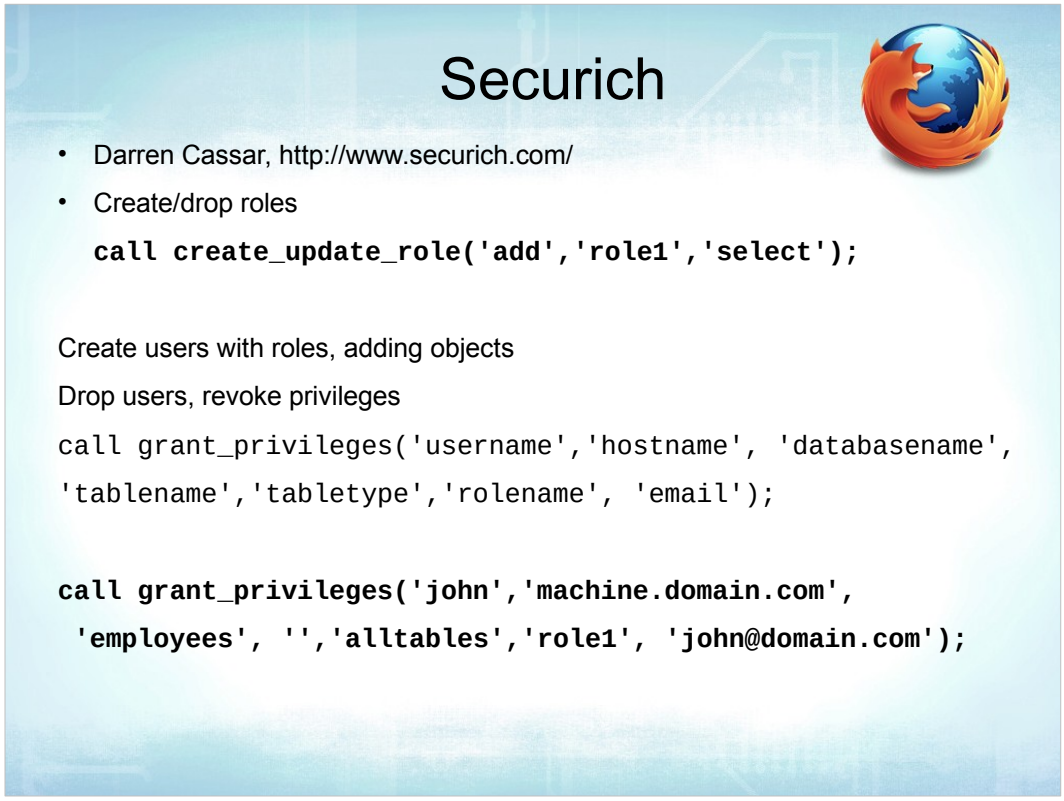
You can also allow people to run commands they otherwise wouldn't by using stored procedures/functions, and you can allow them to see partial data by using views – a view definition is a SELECT query, so you can allow people to see certain columns and even certain rows only.

Access from...?

- localhost only, --skip-networking
- firewall
- Who can [attempt to] DOS you?

Changing ACLs

- Who changes ACLs?
- How are ACL changes audited?
- When do ACL changes happen?



There's a good tutorial too!

create_update_role either creates or updates the role as necessary

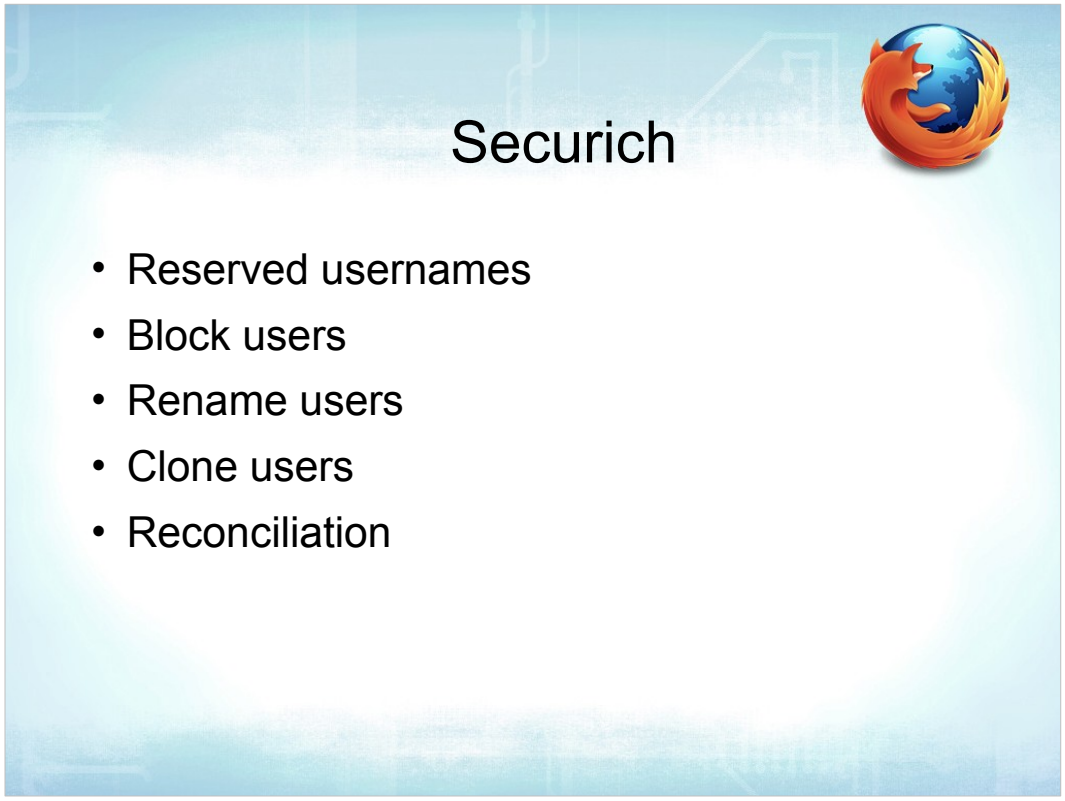
Can on ly drop roles if not in user

Grant privs limitation, if > then truncation happens:

FIELD MAX LENGTH

- username 16
- hostname 60
- databasename 64
- tablename 64
- tabletype 16
- rolename 60
- Emailaddress 50

Tablename can be tblname, regular expression, '' for all, or a stored procedure name



Block - Used to block a particular user, terminating his/her connections if necessary and leave the account around to be unblocked if necessary. This is a useful feature for when a user needs temporary rights.

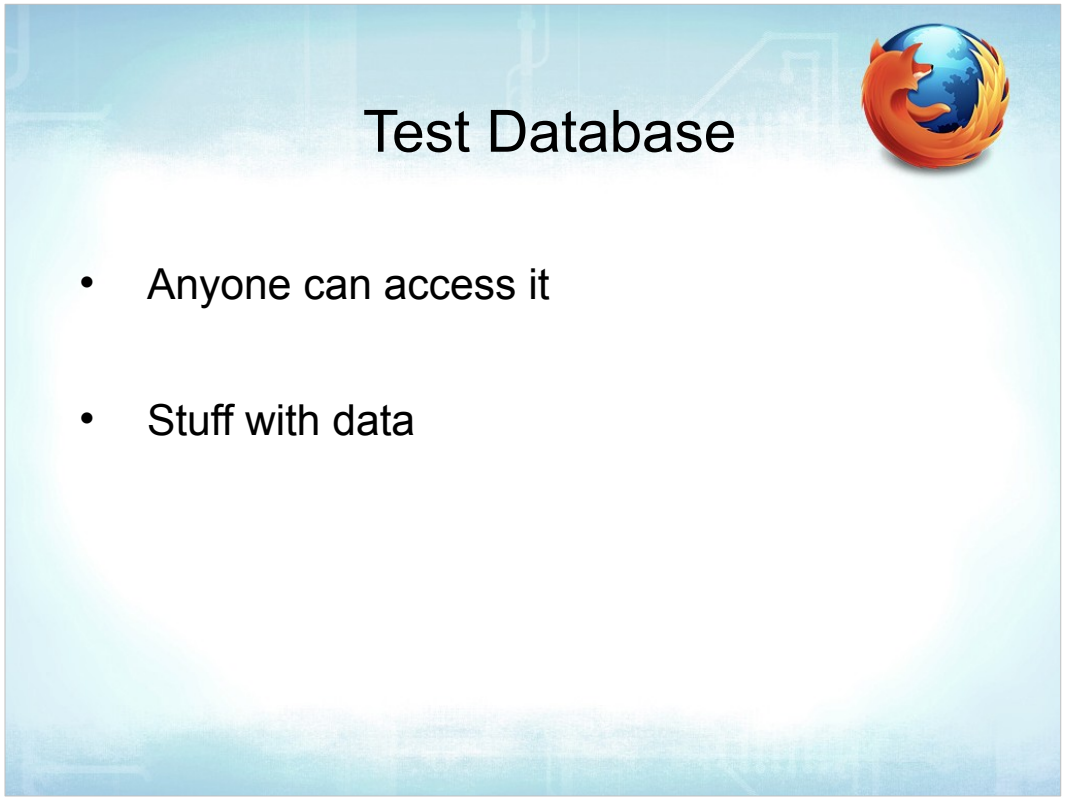
Can reconcile securich's internal db with what's in securich

- password_check();

This is password_check, a procedure used to check for password discrepancies between securich and mysql.

Server Options

- --bind-address
- --skip-name-resolve
- --skip-show-database



Cause DOS!

OS Files and Permissions

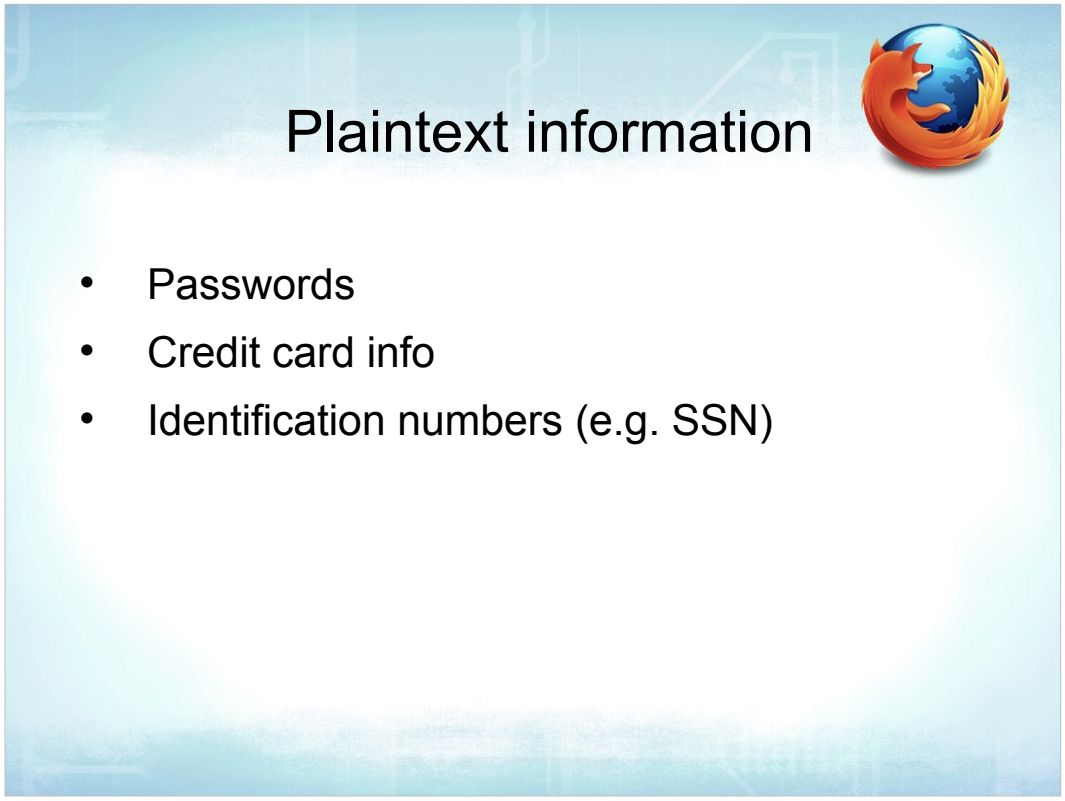
- mysql server user
- mysql server files & logs
- Passwords on commandline
- Office policies/runbook

How Does Your Data Flow?

- Where is data encrypted?
- Where do errors go?
  - Are those logs checked?
- Where does the traffic flow?

Separating Admin Apps

- Same data, different interface
- Performance, e.g. reporting
- Only allowed from VPN?
  - Public vs. easily accessible



App users in mysql db, or app password?

mysql db is in memory, referred to every query. Don't make it too big if you don't have to!

User inputted data into mysql internal table == bad. Imagine html or injection in there...bad.

Can be stolen if db is compromised

How are they transmitted?

Normally (most important)

On reset

What about hash transmittal – if you transmit the hash unencrypted, and others can get to db, they can get to customer.

Users may use them elsewhere

Hashes

- Passwords

*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 = 'password'

- Be careful where encrypted traffic goes!

Where are you encrypting?

The closer to the input source, the better

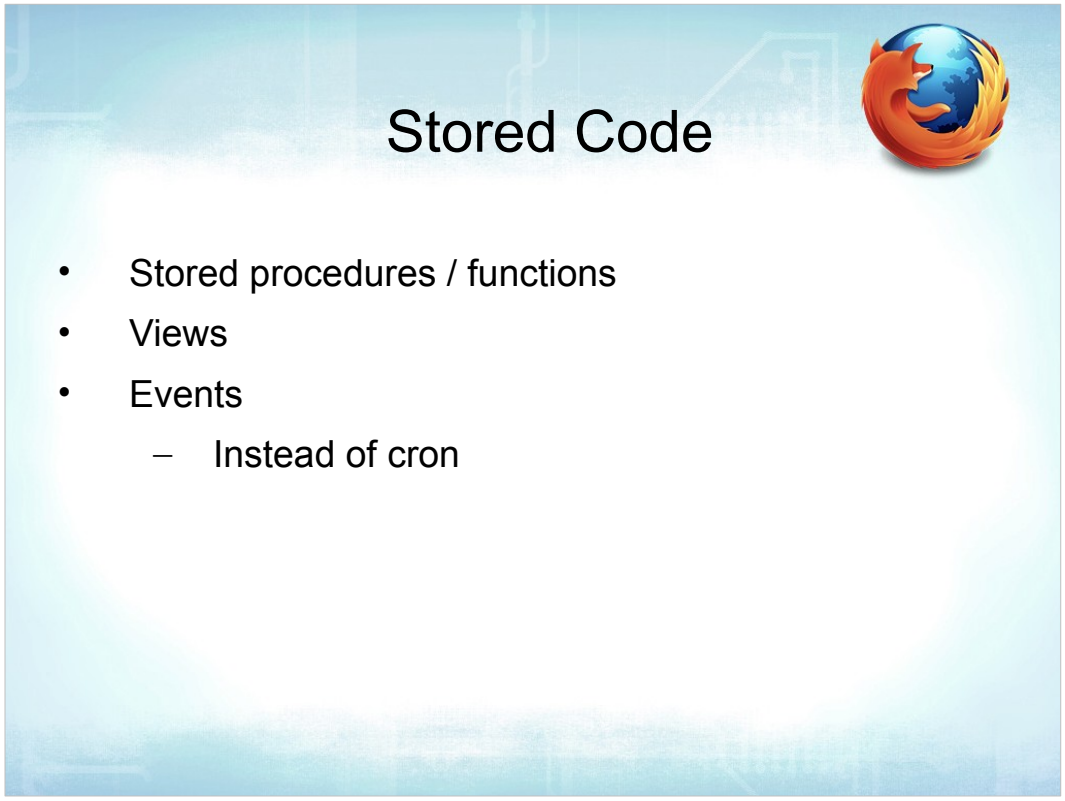
ie, Javascript for HTTP/HTTPS

How are you checking?

Password=hash('foo') ?? hash('foo') then send?

What if I make a web form on MY site that passes info to YOUR site? If checking is only on the page before, there's a problem! Only allow HTTP_REFER from inside...or specific pages.

You can google search for that password hash and find it in Google



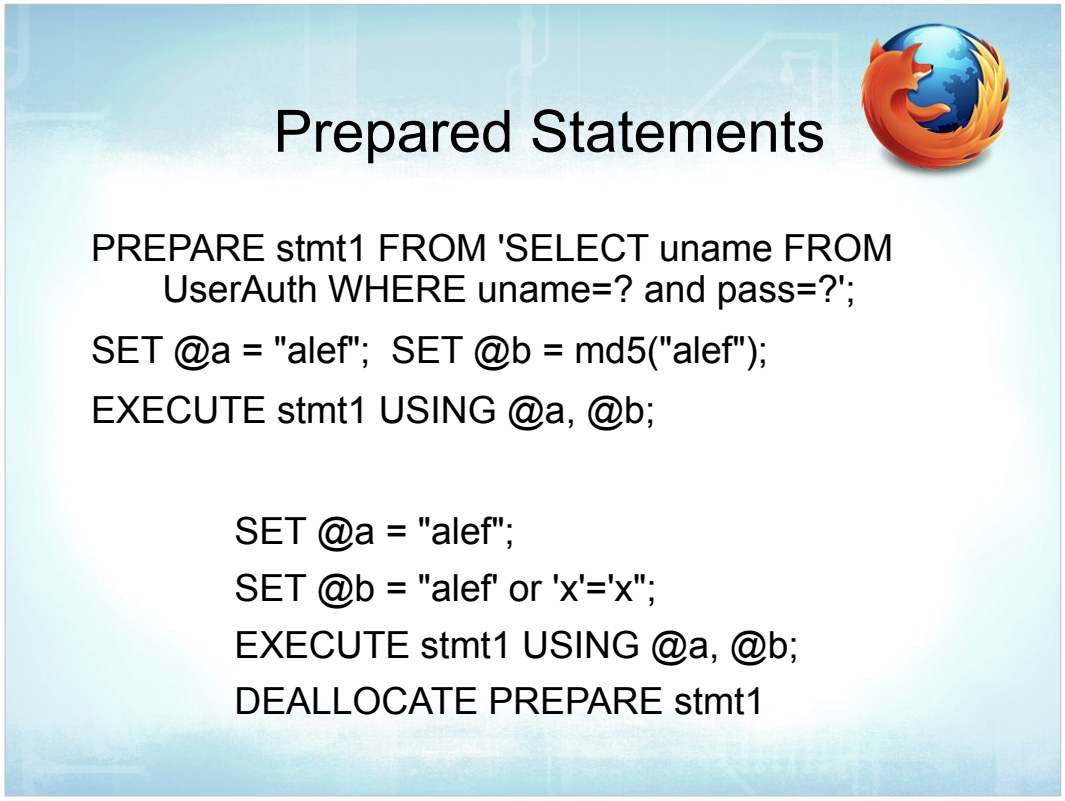
Stored procedures? (MySQL 5)

Can use prepared statements in stored procedures, that's how I do dynamic tables in stored procedures

Prepared Statements

```
PREPARE stmt1 FROM 'SELECT uname FROM
    UserAuth WHERE uname=? and pass=?';
SET @a = "alef";  SET @b = md5("alef");
EXECUTE stmt1 USING @a, @b;
```

Slow! Caches once per SESSION.



Stored procedures? (MySQL 5)

Can use prepared statements in stored procedures, that's how I do dynamic tables in stored procedures

Prepared Statements
Perl Code

```
$query = $sql->prepare("SELECT uname FROM
UserAuth WHERE uname = ? AND pass = ?");
$query->execute($uname, $pass);
```

Prepared Statements PHP Code

```
$stmt = $mysqli->prepare("SELECT uname
  FROM
   UserAuth WHERE uname = ? AND pass = ?");
$stmt->bind_param($uname, $pass);
$stmt->execute();
```

Prepared Statements Java Code

```
PreparedStatement pstmt =
  con.prepareStatement("SELECT uname
  FROM
  UserAuth WHERE uname = ? AND pass
= ?");
pstmt.setString(uname, pass);
ResultSet rset = pstmt.executeQuery();
```

Prepared Statements .NET/C# Code

```
using(SqlCommand cmd = new SqlCommand("SELECT
  uname FROM UserAuth WHERE uname = @uname
  AND pass = @upass",con)) {
      cmd.Parameters.AddWithValue("@userName",
        userName);
      cmd.Parameters.AddWithValue("@pass", pass);
          using( SqlDataReader rdr =
        cmd.ExecuteReader() ){
          ...}
       }
```

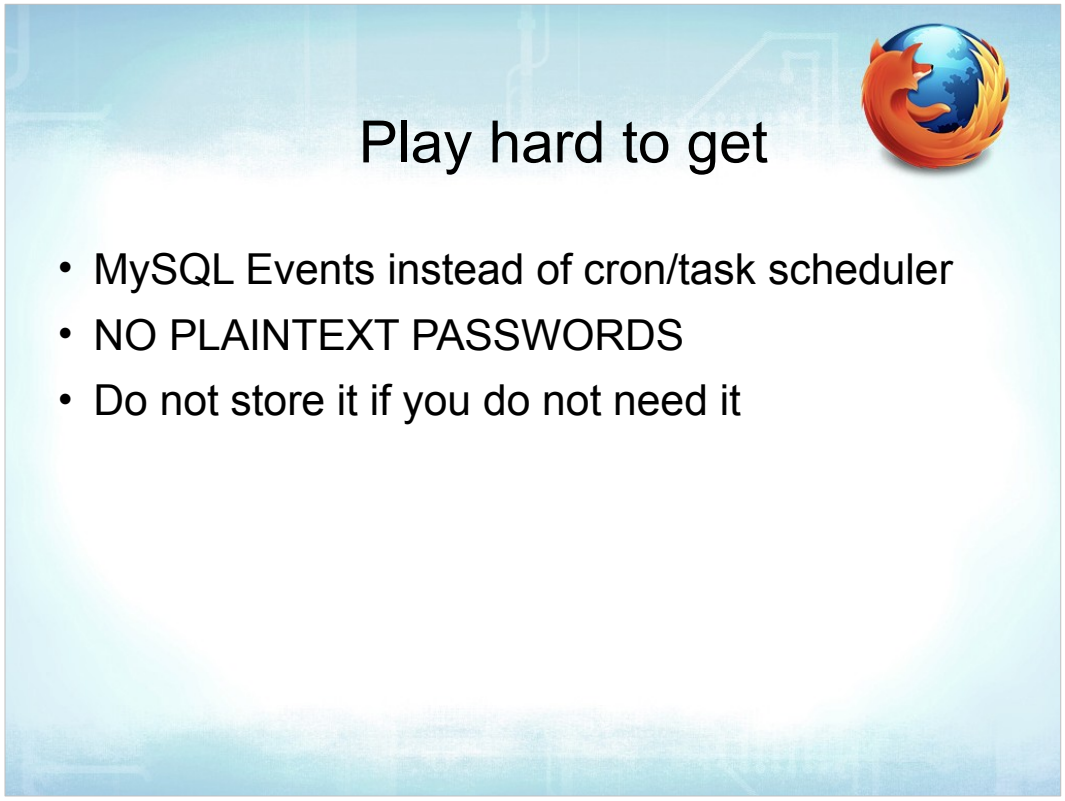
Auditing and Monitoring

- Prevention is one part of security
- Auditing - review and assess security
- Monitoring – alerting of security issues

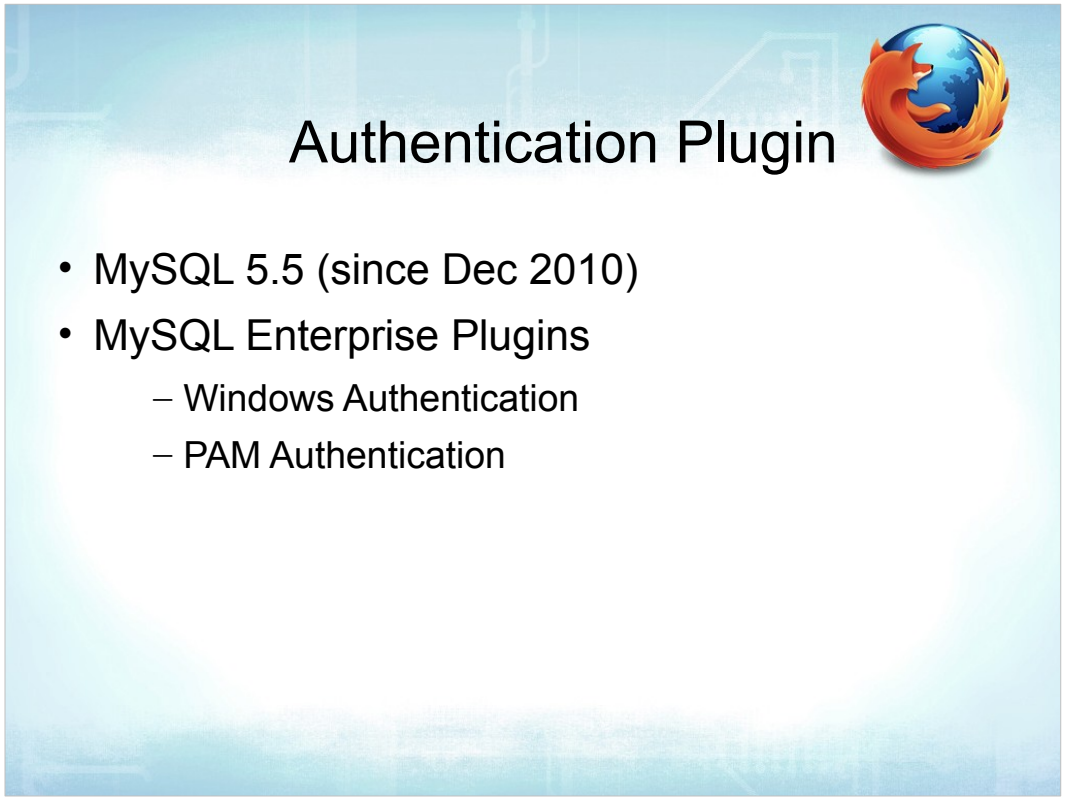
Auditing and Monitoring

- General log to see all login attempts
- Locking out accounts with max_connect_errrors
  - global

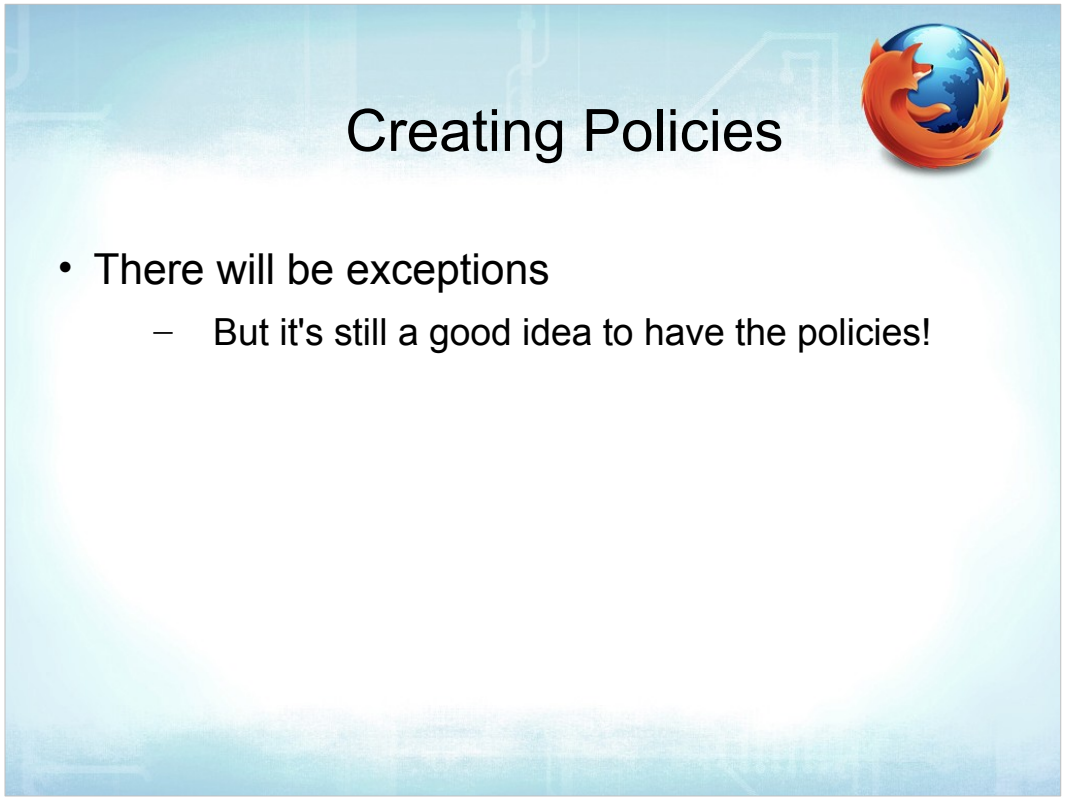
Flush hosts!



No need to store passwords in cron if you use MySQL events. Bonus – it's backed up with the database!



So far these are the only ones, none other yet, but we could use Kerberos auth.



Personal accounts vs. role accounts, how often are each of those passwords changed? When ppl leave? Sometimes it's hard to change app passwords.

Encrypted connections/ replication?

Questions? Comments?

OurSQL podcast

- www.oursql.com

MySQL Administrator's Bible

- tinyurl.com/mysqlbible

kimtag.com/mysql

planet.mysql.com