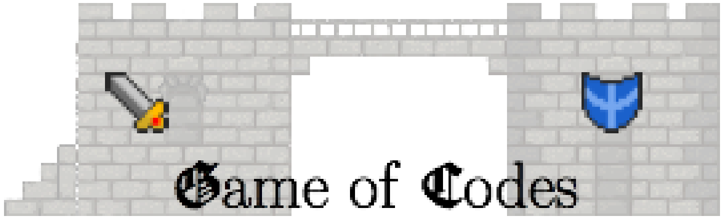
Secure Coding Project



Phase 3: Fixing vulnerabilities &
Development of new features

Fixing vulnerabilities and features

Your client “the bank” wants you to:

- **Fix all the vulnerabilities** that were found by your team and the team that tested your application in phase 2 (starting with the critical vulnerabilities)

- **Fix or add the following features** if you did not have them already:
  - Transfers should be allowed only to existing accounts
  - Clients should see their account number(s) and balance
  - Transaction history entries (visible to both clients and employees) must include: source name, source account, destination name, destination account, amount, description and date
  - Employees must be able to initialize the account balance of customers
  - Batch transfers should allow multiple transfers via the same uploaded file
  - And all other non-working use-cases from phase 1

Development of new features

- First, the bank wants a password recovery via e-mail because many customers complained that they forgot their passwords and could not recover it

- Secondly, the bank wants to increase Internet-banking security, because of some recent news of attacks in the Internet banking systems which send TANs via e-mail. Therefore, instead of sending TANs in plaintext via e-mail, you must **send the TANs in a password protected PDF**.

- Thirdly, clients have the option of using a **personalized Smart-Card-Simulator (SCS):**
  - The SCS program must be implemented in the Java programming language.
  - Whenever a client wants to transfer an amount of money from his/her account s/he must use the SCS.

- NOTE: When a client registers on the bank's website s/he can choose between:
  1. TANs sent via e-mail or
  2. **Downloading a personalized Smart-Card-Simulator (SCS)** program from the Internet-banking website. To use this SCS the client also needs to receive a 6 digit (numeric) PIN. It is up to you to decide what is the most secure and reliable way to electronically communicate this PIN to the client.

Smart-Card-Simulator (SCS) details

- The SCS must have a simple Java GUI for the following inputs:
  - the client PIN number (**not the password** to the Internet banking website)
  - the sum of money to transfer
  - the target account, which the sum will be transferred to
  - a file path for generating a TAN for a given transaction file (one single TAN for multiple transactions)

- The SCS outputs the a unique TAN, which must be copy-pasted by the client on the banking web-site to complete the **transfer via HTML form or batch file**

- The SCS **must not** communicate with the banking web-server or the MySQL database directly. Using the SCS from another machine where the web-server and the MySQL database are not running, should be possible.

NOTE: This assignment requires you to also adjust the server-side of the web-application you developed in Phase 1 of the project, such that is can validate the transaction code output by the SCS.

Deliverables for Phase 3

- Deadline is (Tuesday) 8th December 2015 at 10:10 AM

  → You must hand in via USB stick before this date at the office MI 01.11.040 or on that date in the classroom (lecture will start at 10:10 AM)

  → There will be a 1% grade penalty for every minute after the deadline

- USB stick must contain the following files and folders:

  1. **SamuraiWTF-TeamX-Phase3.ova** (where X is your team number)
  2. Folder called **Credentials-TeamX-Phase3** contents of this folder explained on next slide
  3. **OWASP-Checklist-TeamX-Phase2.xls (Excel 97-2003)**
  4. **SourceAndBinary-TeamX-Phase3.zip** (containing all PHP, MySQL sources and the C/C++ binary for batch processing, **NOT the C/C++ source code**)
  5. **Presentation-TeamX-Phase3.pdf** (NO .PPT or .pages files)
  6. Folder/directory called **Videos-TeamX-Phase3** containing video demos for each newly implemented functionality

     The structure of the presentation and videos are presented on next slides

Structure of Credentials Folder

The Credentials-TeamX-Phase3 should contain the following files:

- Non-password protected PDF files called **<username>.pdf** with the TAN numbers of existing users, which must have a transfer history and money in their accounts, where <username> indicates the username of the account to which the TANs belongs to.
- **general-info.txt** file should contain the following information:
  - the username and password of the OS user
  - the username and password of the MySQL database
  - with the username and passwords of all the existing users.
  - the username and password of the admin/employee user for your web-application
  - the URL to access your web application
  - the location and structure of the folder(s) where you have the source files of your web application.
  - the name and location of any third party libraries that you have used in phase 3.

Structure of Phase 3 Checklist

The checklist will not be presented during the lecture

- Custom version of the OWASP testing checklist https://docs.google.com/spreadsheets/d/1RZYzw7OXjjmiG0U1F8gYCV_SGrZY3guSncRwluwmiGM/edit?usp=sharing

- Copy column E from OWASP checklist of the team that tested your application in phase 2

- Fill out column F with the fixes that you performed in phase 3. Add notes with references to slides from the Presentation-TeamX-Phase3.pdf

- See the description of tests in the OWASP Testing Guide https://www.owasp.org/index.php/OWASP_Testing_Project

Structure of Phase 3 Presentation

Must be presented on **Friday**, December 11th 2015 (from our laptop) **Time: 6 Minutes**

1. First slide: team number and team members (full-names)

2. Second slide: enumeration of use-cases that were **fixed / work / do not work**. At least the following use-cases should be there:
   a) All use-cases of phase 1 (see Phase 1 Deliverables document)
   b) Password recovery
   c) Encrypted TAN generation and delivery via e-mail to customer
   d) Download of SCS after registration
   e) Transfer using SCS

3. Core of presentation: at this point in the presentation you must talk about:
   - **Vulnerabilities** (possible exploits of the vulnerabilities) **and how you fixed them**.
   - Feel free to go into technical details, but keep it **short, clear and entertaining**.
   - Report possible changes in OWASP and CVSS values (e.g. from high to medium, etc.) **including reasoning for these changes in the values.**

Structure of Phase 3 Presentation (continued)

4. Fix slides: For each vulnerability discovered in Phase 2 by your team or the team that tested your application, you should summarize the fix that you implemented:
   - Path(s) of file(s) which was modified for the fix
   - Line number(s) which were modified in each file
   - NOTE: You can use a diff utility but you should only report the changes relevant for one particular fix per vulnerability (not just a diff of all changes to phase 1).
   - Textual description of your countermeasures and the reason why they fix the problem.

5. Use-case slides: will serve as a user manual describing all the use-cases that work using the same format as in the Phase 1 Deliverables document for each use-case.
   - Mention the format of the batch file and any other data formats needed to use the application!
   - Each use case must have a video (screencast) associate with it that demonstrates the main-flow of the use-case. **A user must be able to use the app by simply looking at the video.**
   - The video must have the same name as the use-case and a resolution of **1280x1024**.
   - The format of the video must be MP4, the video codec must be H.264 and the audio codec must be AAC. The file must be located in the **Videos-TeamX-Phase3** folder.
   - **The quality of the video must be good enough to be able to read the text shown on the screen.**
   - **In case passwords are typed in please add video annotations for the viewer to know the password.**

6. Time tracking table (info / student / task at hour granularity)
   Tasks should be evenly distributed – project management part (this will be graded)