NETW7008 (Secure Programming)   
Coursework 2022

# Summary

The software development is for the 30% of your module mark. You will produce a secure prototype of a job recruitment website in the programming language C/C++, accessed through a web interface. This will test the following module level outcomes.

* Have a thorough understanding of the main attack vectors commonly used to attack software and be able to design and implement software that reduces the likelihood of those vulnerabilities being exploitable
* Design and implement concurrent and distributed software which operates in hostile environments
* Design and implement secure software that utilises the underlying security model of the OS and hardware.

# Submission

The software development must be uploaded as a zip file (for the code) and a single docx or pdf file (for the report) before Friday Week 12, April 29th 1pm. A demonstration will be required and further information about that will be released closer to the time. The University rules concerning plagiarism, collusion and contract cheating apply.

CW2: Software Development

The software you are writing is a CGI program written in C/C++. This means that it is running on a web server and it will be accessed through a web interface. The software will be a job recruitment portal with similar functionality to the program you examined for CW1. However, it must be a CGI program and a submission consisting of minor modifications to the code you have already been given for CW1 will not be accepted.

Here are the functional and non-functional requirements for the software development. You will be marked on your understanding of potential attack vectors against secure software and on the principles of designing secure software, and on how well you design and implement software that reduces the likelihood of those vulnerabilities being exploitable.

# Functional Requirements

|  |  |
| --- | --- |
| FR1 | There are three kinds of users: companies, applicants and a single administrator. All of them can register an account and set a password. |
| FR2 | Each company can register the skill it requires, subject to admin approval. They can also search for approved applicants who have that skill. These two abilities should be offered as two separate operations. |
| FR3 | Each applicant can register the skill they have, subject to admin approval. They can also search for approved companies who require that skill. These two abilities should be offered as two separate operations. |
| FR4 | The Administrator can view a list of companies for approval and a list of applicants for approval. They can approve or reject each applicant and each company or leave the decision on each one for later. There is only one Administrator account. |
| FR5 | The process of logging in should use two-factor authentication. The user must enter a second password sent by email after the main password has been entered. The email address to be used is the one entered when registering the account. If you are not able to install the relevant mail library, you can simulate the process of emailing by appending to a “mail spool” text file representing all the emails that have been sent. |
| FR6 | The Administrator account, in addition to the protections of FR5, must also be authenticated by a “hardware” token, which should be implemented as a piece of challenge-response software. |

# Non-Functional Requirements

|  |  |
| --- | --- |
| NFR1 | You may use your own web server running on your own machine if you wish. However, the department has provided the SOTS server, which you can use instead. Your login details will be the same username and password you use for Moodle and Google Suite. |
| NFR2 | The system must be developed in C/C++ and must use CGI to interact with the web pages. You may use the C/C++ CGI libraries, which have been installed on SOTS, if you are using SOTS. Here is one of many tutorials on them:  <https://www.tutorialspoint.com/cplusplus/cpp_web_programming.htm>  Similarly, you can use libraries just as the MySQL Connectors for C or C++. Both these are the CGI libraries are covered extensively in the appropriate practical sessions. |
| NFR3 | The system must be robust and secure. Specifically, it should be capable of mitigating many kinds of attacks covered in the module, as detailed in the marking scheme. SSL must not be the sole means of preventing these attacks. |
| NFR4 | The system must be designed with maintainability, security and reliability in mind and according to best practice in designing and implementing secure software. Defensive software practices should be used throughout |
| NFR5 | Your code should be commented and have sensible and consistent naming |
| NFR6 | The system should be responsive and easy to use |
| NFR7 | You may use cryptographic libraries if you wish. |
| NFR8 | Your report must explain why you believe you have satisfied NFR3, NFR4, NFR6. |
| NFR9 | Your report must explain why you believe you have satisfied FR1, FR2, FR3, FR4, FR5, FR6. |

# Marking Scheme

The marks pertaining to the functional and non-functional requirements will be awarded by examining your code including comments, your demonstration and your report, which must document the implementation process and make use of the following headings. You will lose the opportunity to acquire marks if you miss out requirements or do not provide the other documentation asked for.

* Documented Implementation of Functional Requirements - 35% (broken down as follows)
  + Account registration (FR1) – 5%
  + Company / Applicant functionality (FR2/FR3) – 10%
  + Administrator functionality (FR4) – 10%
  + Hardware token (FR5) – 5%
  + Two-factor authentication (FR6) – 5%
* Documented Implementation of Non-functional Requirements – 30% (broken down as follows)
  + Session handling using cookies (NFR3, NFR4) – 10%
  + Redirection of users not logged in (NFR3, NFR4) – 5%
  + Input sanitization (NFR3, NFR4) – 10%
  + Encryption of stored information (NFR3, NFR4) – 5%
* Other documentation – 35% (broken down as follows)
  + Documentation of system design including demonstration - 10%
  + Justification of system design - 5%
  + Testing and Security Audit – 10%
  + Conclusion, reflection and references – 10%

We are expecting a typical submission to be about 1000 words (excluding headings, references and code) but you will not be penalised for exceeding this. Where you need to refer to code in our documentation, make sure you include the appropriate code extracts.