Name: Joshua Bonsink
Email Address: josbonsink@gmail.com
Instant messenger name and protocol (if any):

1. Skype: jos.bonsink
2. IRC: JBonsink

Nationality: Dutch
Principal spoken language: Dutch/English

1. Top project choice (can be one of our project suggestions or your own)
   I have chosen project 8: Integrated Malware Simulator and Emulator.

2. Are you willing and able to work on other projects instead?
   Yes, but I prefer project 8 since I have invested a lot of time preparing. It is my favourite.

3. Please describe you preferred coding languages and experience:
   Comparing to my classmates I started late with programming. I wrote my first program in Assembly in September 2011. Since that time my knowledge has grown. Please find the languages that I refer to below.

   (a) C++: I have written a couple of simulation programs during the course "Simulation and Modelling". One of those programs was a forest fire simulator which I wrote with a fellow student. The final assignment during the course "Graphics & Game Technology" was to build a 2D physics game using the Box2D Physics engine.

   (b) Python: I have written a lot of small programs with Python during the Image Editing" and "Networking & System security" courses. I have also worked on web server based on Flask. It was a complex piece of software purely written in Python.

   (c) C: The most experience I acquired with C was during the "Data structures" course. I learned the relevant techniques such as pointers. A lot of the assignments given at the University are written in C. Although I have worked a lot with C, I do prefer C++.

   (d) Java: The first time I wrote a program with Java was during the "Java" course. I learned how to implement the Object Oriented programming paradigm. I also wrote a sidescroller game with Processing which is based on Java.

   (e) HTML/CSS/Javascript: I have build a webshop for a virtual game retailer. I also worked on a plugin for Sakai, a website that increases the interaction between teacher and students. The teacher submits a question, the website shows the question to the students and students answer. Then the website would automatically score their answers and present the results to the teacher.

The coding languages are listed in order. My favourite language is listed at the top.

4. Please describe any Windows, Unix or Mac OS X development experience relevant to your chosen project.
   I have acquired relevant experience for project 8 during the "Networking & System security"course at the University of Amsterdam. The following course assignments are written in Phyton.

   (a) TCP socket programming: send an email with the use of TCP client socket programming.

   (b) Client-Server architecture: multiplex sockets to handle multiple simultaneous connections and combine socket I/O with other operations. The goal was to create a network of simulated sensors.

5. Please describe any previous usage with Honeynet Project tools or honeypots in general.
   I have seen the two demo videos that were available for the IMALSE project. Currently I am able to execute the program successfully. However, I have not used it for scientific research yet.

6. Please describe any previous Honeynet Project or honeypot related development experience, including details of any patches, code or ideas you may have previously submitted.
   It was early April that I first heard of Honeynet Project. This also means that it would be first time that I contribute.

7. Please describe any previous Open Source development experience, including projects you have worked on
   I worked on the Open Source Project Sakai during Project Software Engineering course. In collaboration with my classmates, a team of 15 people, I have developed a plugin that stimulates the interaction between teacher and student during lectures.

8. What school do you attend and what is your speciality/major at the school?
   I am a Computer Science undergraduate from the University of Amsterdam.

9. How many years have you attended there?
   I have attended the University of Amsterdam for two years now.

10. What city/country will you be spending this summer in?
    I will be spending this summer in The Netherlands.

11. How much time do you expect to have for this project?
    I expect to have at least 40 hours per week available for this project.

12. Please list jobs, summer classes, and/or vacations that you'll need to work around.
    I am not planning to go on holiday. I have a par-time job that takes about 10-15 hours per week of my time. I still have a course that ends on June 28th. The first semester of my third year starts in the first week of September.

13. Have you participated in any previous Summer of Code projects? If so please describe your projects and experience, including what you liked or didn't like about the experience.
    No, but I am enthusiastic and hoping that this will be the first time.

14. Have you applied for (or intend to apply for) any other Google Summer of Code 2013 projects? If so, which ones?
    I have not and do not intend to apply for other projects. I am focusing on this project.

15. If you have a URL for your resume/CV, please list it here.
    You can view my resume on Linkedin:
    `nl.linkedin.com/pub/joshua-bonsink/46/a5a/327`

16. If you wish to list any personal/blog URLs, do so here.

17. Please describe your proposed project in detail, including deliverables and expected time line with milestones (this is the long answer, so spend most time here!).

    Please find my ideas to improve the existing malware simulator and emulator below. They are listed in the order of processing. I am able to work full-time on this project from week 27 through week 35.

    (a) Refactor the code:
        The existing code lacks documentation and the relationship between some of the objects are obscure. I propose to start the project with refactoring some of the existing code. In order to bring the code more to a state of "self documenting code", I suggest to add some additional functions and more commenting where necessary.

        I want to remove al the "import *" because one can not see directly where the imported methods are defined. Import * is not a good idea to use in production code. When two functions with the same name are imported, you are unable to determine which one is used.

        The code responsible for the simulation can especially use some commenting. Firstly the experiment class is created, which can for example be TopoExperiment. TopoExeriment is the child of the mode (e.g. ImalsePureExperiment) and the mode is again a child of ImalseExperiment. Some documentation should clarify these kinds relationships.

I am planning to perform these actions in weeks 26.
Milestones:

i. Refactor and document the code responsible for the emulation
ii. Refactor and document the code responsible for the different types of simulations

After finishing, I should have an even better understanding of all the inner workings of IMALSE.

(b) Background traffic generator:
Network Traffic Models can capture the characteristics of actual network loads. There is no single traffic model that can efficiently capture the traffic characteristics of all types of networks, under every possible circumstance. Each model is best at capturing different traffic characteristics.

We can generate background traffic with the use of these kinds of models. The current background traffic generator is very simple. It is situated in *core/configure*. It will generate a DOT file, which defines the network topology and how the background traffic should look like. This DOT file is only used when it is specified during a ComplexNetExperiment. Without specifying this or when other experiments are performed, normal nodes will not generate traffic.

SADIT, another project of Jing Wang, has an better integrated flow data generator which has some network traffic models implemented, such as the Markov models. This part of the code can be extracted and relatively easy integrated into IMALSE. As long as the format of the DOT file remains consistent, the change of code will have little influence on the rest of IMALSE.

I am planning to implement this idea in week 28. Milestones:

i. Extract the code from SADIT
ii. Integrate into IMALSE
iii. Test and clean-up code

(c) Additional scenarios:

i. Spamming: With the help of a botnet existing out of thousands of bots, an attacker is able to send massive amounts of spam. The spamming process can be carried out in collaboration between bots performing different tasks, some working as spam harvesters while others as spam generators. Some bots act as content servers while others as SMTP servers.
ii. Sniffing Traffic: Bots can also use a packet sniffer to watch for interesting clear-text data passing by a compromised machine. The sniffers are mostly used to retrieve sensitive information like usernames and passwords.
iii. Keylogging: If the compromised machine uses encrypted communication channels, then just sniffing the network packets on the victim's computer is useless since the appropriate key to decrypt the packets is missing. With

the help of a keylogger it is very easy for an attack to retrieve sensitive information.

iv. Spreading new malware: Botnets are often used to spread new bots. All bots implement mechanisms to download and execute a file via HTTP or FTP. Since botnets can also have spamming capabilities it is also possible to spread an email virus. A botnet with 10.000 hosts which acts as the start base for the mail virus allows very fast spreading and thus causes more harm.

I'm planning to implement these additional scenarios in weeks 29-32, one week for each additional scenario. However, if the time frame is too tight, the number of the additional scenarios will be reduced.

(d) Unify the two GUI systems:
Create a user friendly GUI. Instead of just unifying the two GUI systems, wouldn't it be great if there was one GUI for IMALSE. Where you can choose simulation or emulation and input all the necessary settings. This would minimize the use of the terminal to execute the different programs.

I am planning to create a GUI with pyQt, a Python binding of the cross-platform GUI toolkit Qt. The current GUI is written in TCL which is kind of old-fashioned. A modern looking GUI can be created with pyQt and pyQt is more programmer friendly. NetAnim may be used to animate the simulation in IMALSE.

I am planning to implement this idea in weeks 33-36. I am aware that a delicate GUI will take longer than a simple one. So, if the time frame is too tight, I will create a more basic GUI. Milestones:

i. Create GUI elements to start the emulation mode.
ii. Create GUI elements to start the simulation mode.
iii. Create GUI elements to draw the network topology.
iv. Create GUI elements to visualize the topology and traffic with NetAnim.
v. Test and clean-up code.

(e) Final clean-up code and finish documentation:
I'm planning to do this in weeks 37. My workload will probably be minimal during the beginning of my third year thus my working pace will only be slightly decreased.

18. Why are you well suited to perform this project and why are you interested in it?
I'm passionate about things that interest me, especially this project. I often think about IMALSE, trying to come up with ways to improve it. I have often worked with Python and I am confident that I can handle the complexity of this project. I am looking forward to the end of June when I can finally focus all my attention on improving IMALSE.

I am going to specialize in security after finishing my Computer Science bachelor. This project will give me the opportunity to increase my knowledge in the field of security. It perfectly suits my future career.

19. Have you been in contact with us previously about your project ideas on IRC, and if so, what is your IRC nick?
    I have been active on IRC and my nick is JBonsink. Further I have contacted Jing Conan Wang by email to discuss my project ideas.

20. Have any of our members met you face to face, such as at one of our recent public events (Paris or San Francisco Bay Area or Dubai)? If so, please list who/where?

    No, I have only met members of Honeynet on IRC.