Hackmi2: Threat Modelling in Social Networks using attack centric model.

Computer Science Honours Project Report

By: Molulaqhooa Linda Maoyi

Supervisor: Dr Anne Kayem

		Min	Max	Chosen
1	Requirement Analysis and Design	0	20	15
2	Theoretical Analysis	0	25	15
3	Experiment Design and Execution	0	20	5
4	System Development and Implementation	0	15	10
5	Results, Findings and Conclusion	10	20	5
6	Aim Formulation and Background Work	10	15	10
7	Quality of report Writing and Presentation	10		10
8	Adherence to Project Proposal and Quality of Deliverables	10		10
9	Overall General Project Evaluation	0	10	0
Total marks		**80**		**80**

Department of Computer Science

University of Cape Town

2012

ABSTRACT

Social networks have become ubiquitous nowadays as one of the most used internet services. As with every new web technology, they are prone to a plethora of vulnerabilities that might compromise on the security of the system and lead to privacy concerns such as identity theft. In addition to these concerns, there are other vulnerabilities that exist that might be potentially dangerous to the overall security of the social network. In this project, we develop a social network as a case study in order to aid us in the study of the threat modeling approach so we understand how to better model threats and vulnerabilities in a system. This is important because many systems have assets of value such as user passwords that are protected. However, there might be vulnerabilities associated with these assets and therefore ,in this project, the threat modelling approach looks at the security vulnerabilities from the attackers view in order to understand how to design better security mechanisms and countermeasures.

For the threat modeling process, we will be using the Microsoft Threat and Analysis Modeling (TAM) tool, Microsoft Software Development Life Cycle (SDL) and the SensePost Corporate Threat Modeling (CTM) tool. Our aim in this project is to make a comparison on the tools and how they model and generate threats.

ACKNOWLEDGEMENTS

Foremost I would like to express my gratitude to my Supervisor, Dr Anne Kayem for her great support during this project, for her patience, motivation and guidance throughout the writing of this thesis. I would not have asked for a better Supervisor.

My gratitude also goes out to Dominic White from SensePost for the material and advice he supplied us during the course of this project.

I would also like to thank Prof James Gain for his constructive critique during the early stages of this work.

I would also like to thank my project partners Rotondwa Ratshidaho and Sanele Macanda for the many times I was clueless on what to do and they shed some light, you guys are awesome!

Lastly I would like to express my deepest gratitude from the bottom of my heart to my beloved parents and my siblings. Thank you for all the continuous support you have given me all these years. I wouldn't be where I am today without you.

To my Father, Israel Maoyi and Mother Tsholofelo Maoyi

~God bless you all~

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROBLEM OUTLINE

Social networks have become a ubiquitous and popular internet service. As with every new web technology, they are prone to security vulnerabilities that might compromise the security of the system and lead to privacy violations such as identity theft[1]. This is because many systems are deployed with security mechanisms that have not been checked as rigorously as the rest of the system. Scandals such as the one that occurred in July of 2011, where a security consultant (Ron Bowles) used a piece of code to collect personal data off Facebook and published it on Pirate Bay[2], only serve to emphasize the need for more thorough security and privacy verifications [1].

Threat modeling is description of a set of security aspects and is useful in applying a structured approach to system security design. With threat modeling, one can identify and categorize potential security vulnerabilities in a system and then develop countermeasures accordingly. Approaches to threat modeling include:

- The attack centric model which is centred on the attacker's goals and motivations for breaking into a system. In this case, we want to evaluate how an attacker would achieve this and why. The reason is because whether we like it or not hackers do exist and understanding how an attacker might invade a system will aid in protecting it.

- The software centric model is aimed at detecting vulnerabilities in the system's design. In this case, we step through the different components of the system and look for vulnerabilities that an attacker might exploit.

Having a blueprint of potential threats is useful; from the system designer's view point, because potential attack scenarios as well as the countermeasures can be simulated before the application is deployed. However, until recently, not much consideration has been given to comparing threat

[1] Stealing someone's identity and using it without their consent.
[2] A popular file sharing website.

Modeling tools and evaluating their capability in dealing with threats.

This project aims to investigate and compare threat modeling tools in order to evaluate their behaviours in finding potential threats on a social network. Molulaqhooa will be analysing the social network from the attacker's point of view using the Microsoft TAM (Threat and Analysis Modeling Tool); Rotondwa will also be impersonating an attacker using the Microsoft SDL (Software Development Cycle) Threat Modeling tool and Sanele will take the role of the Social network system administrator using the SensePost CTM (Corporate Threat Model).

1.2 RESEARCH QUESTIONS

The main research questions of this project fall under the following categories:

- What are the social network security risks?

  Attackers can use many paths in a social network to exploit vulnerabilities that can be used to do harm to businesses, organizations or users. Each of these paths represents a risk that may, or may not be serious enough to warrant attention and so we need to identify and categorize the risks in terms of importance or danger [2].

- What sort of attacks on social networks can be exploited by malicious code and how does one model these attacks in order to mitigate them?

- What counter measures could be employed by the threat modeling tool in identifying threats and mitigate them?

- For the same social network, how do the Microsoft TAM, Microsoft SDL and SensePost CTM threat models differ with respect to identifying threats and providing mitigation strategies?

1.3 PROPOSED SOLUTION

A social network will be developed using the Elgg open source social network engine where users can perform basic social networking functions such as registering, logging in, creating profiles, updating statuses, blogging and chatting. This social network will serve as a test bed for evaluating the threat modeling tools and provide countermeasures for potential attacks. The reason for developing our own social network is because the nature of this project requires the use of malicious code to exploit vulnerabilities detected by the threat modeling tool. Malicious code can take form of SQL injections attacks where an attacker inputs harmful SQL commands in a web form entry field in an attempt to pass the commands to the database, for example, deleting all users from the database or stealing user passwords. For this reason, there might be legal issues involved should the attacks we carry out succeed.

Molulaqhooa and Rotondwa will use the threats generated by their threat modeling tools and try to exploit them. After the exploit, each will write an antidote to serve as patch for the exploited vulnerability. Molulaqhooa will be particularly focused on SQL injections, XSS (cross-site scripting) attacks and Denial of Service attacks which will be discussed in Chapter 9. Sanele, as the social network administrator will be focussed on using the SensePost modeling tool in identifying attacks by stepping through the different components of the system and find the weaknesses that an attacker might exploit.

1.4 KEY SUCCESS FACTORS

The key success factor in this project will be making sure that the social network is secure from potential attacks the threat modeling tool exposed. This, however, will not imply that the social network will be 100 % vulnerability free, but instead, it will be more secure than it was before.

1.5 REPORT OUTLINE

In Chapter 2, relevant literature pertaining to web security and threat modeling will be discussed. Chapter 3 will be looking at the Elgg frame work and why it was chosen for this project. Chapter 4 describes Hackmi2. Chapter 5 covers the theory on threat modeling and how it is used to model vulnerabilities in applications. Chapter 6 describes the analysis of the hackmi2 social network using the Microsoft TAM tool. In Chapter 7 we compare threat models. Chapter 8 looks at the vulnerabilities found in the social network and Chapter 9 Contains the overall conclusion.

CHAPTER 2

BACKGROUND

2.1 WEB APPLICATION SECURITY

Web applications provide a client interface to end-users through web pages in order to access server functionality. These pages tend to have scripts that are dynamically executed by the client web browser. According to E.Chien [3], most web applications enforce simple security policies such as, for web-based email, preventing scripts to execute in untrusted email messages. Although this seems like a good approach, these applications are currently prone to a plethora of attacks such as SQL-injections, cross-site scripting, cookie theft, browser hijacking and unvalidated redirects and forwards [4]. These vulnerabilities allow attackers to have access to applications through bypassing of authentication which in turn allows them to obtain sensitive data like credit card numbers and illegal transfer of funds from commercial websites. According to surveys conducted by the MITRE corporation in collaborating with the U.S Department of Homeland Security [5], security problems in web applications were the most commonly reported vulnerabilities on the internet and J.Walden et al asserts that the number of vulnerabilities discovered each year have increased at an exponential rate since 2000 [6].

2.2 WHY ATTACK WEB APPLICATIONS?

There are numerous motivations for attacking web applications and these have been discussed for years in various books and internet forums. In this section we will look at web application features that makes them attractive to attackers because understanding these factors will aid in gaining deeper insight in what defences can be put in place to mitigate risks. According to J.Scambray et al [7], the following are the most common motives why attackers hack web applications.

- **Ubiquity:** Web applications are everywhere nowadays and continue to increase rapidly across private and public networks. This provides attackers an abundance of "juicy" targets to hack.
- **Anonymity:** The internet has a lot of unaccountable regions and it is relatively easy to launch attacks with little fear of being traced. Attacks in the web usually happen through

open HTTP/S[3] proxies that remain plentiful on the internet. Sophisticated attacker will even route requests through different proxies such as onion routing[4] to make it harder to trace them. This problem remains the primary reason for the increase in malicious hacking since this anonymity strips away one of the primary restraints for such behaviour in the physical world (i.e. being caught and punished).

- **Bypassing Firewalls:** Inbound HTTP/S is permitted by most firewall policies and this is not a vulnerability associated with the firewall but by the administrator's configured policy. The good news for attackers is that this configuration is going to increase in frequency as more applications migrate to HTTP. This is evident in the growing popularity of Social Networking web applications where sharing photos via the web, chatting and personal blogs feature.

- **Custom Code:** With the increase of easily accessible web development applications such as LAMP (Linux, Apache, MySQL and PHP) and ASP.NET, most web applications are developed by programmers with little or no experience of security programming which leaves the applications open to security loopholes.

- **Constant Change:** There are many people who are constantly "touching" a web application. These include developers, system administrators and content managers of all sorts such as a marketing team in a company. Most of these users have no adequate training in security and yet, they are authorized to make changes to a complex web-facing application[5]. This level of dynamism makes it hard to enforce consistent security policies.

- **Money:** Current trends in hacking indicate that the motivations for web hacking have moved from fame and reputation to fortune. There has also been an increase in the number of organized crime enterprises built upon making profit from web hacking. Whether through breaking into servers, fraud directed to users through phishing or extortion through denial of service attacks, the sad reality is that web crime pays.

[3] Hypertext Transfer Protocol Secure (**HTTP/S**) is a popular communications protocol for secure communication over a computer network
[4] Onion routing is a technique for anonymous communication over a computer network
[5] A web application that is designed to be accessed by users or organisations over the internet

2.3 SOCIAL NETWORKING WEB APPLICATIONS

2.3.1 Introduction

Social networks emerged in the 21st century because of the information technology boom and world globalization through the Internet [8]. This started in 1997 with the advent of SixDegrees.com and since then, this has resulted in the establishment of many social networking sites which were based on the "Social-circle model" [9] depicted in Figure 1. The notable ones are MySpace, LinkedIn, YouTube, Twitter and Facebook. These social networks provide users with options of personalizing their profiles where they can post personal data, share information with friends, chat and uploading multimedia information. While these may have many benefits, there are also underlying security threats that can be exploited maliciously by hackers and cybercriminals to compromise the system through several security violations, such as identity theft and information leaking.

Figure 1 : Social-Circle Model

The next section shows the different vulnerabilities exploited by attackers on social networks as pointed out by C. Laorden et al [10].

2.3.2 Weak Spots in Social Networks

The most common vulnerability in social networks is associated with the difficulty of removing all user information when deleting an account. This is due to the licence agreement clauses which appear when a user tries to leave a social network and the rights which were transferred to the Social network when the content was uploaded. Another reason that social networks would keep user data is for in case a user decides to go back and use the social network and does not have to go through the pain of creating a new profile. So in order to delete uploaded contents permanently, a user has to manually delete content one by one.

Weak security methods are a downfall for most web services; some include the lack of restricting the amount of login attempts that a user can perform which leads to brute force attempts. For example, in 2010, a blogging network, Gawker media, was a victim to a hacking attack. The network consists of news site gawker.com, gizmodo.com related to gadgets, jezebel.com related fashion, kotaku.com related to games, deadspin.com related to sports and lifehacker.com [11]. The network had approximately 1.5 million users.Hackers gained unauthorized access to the user's databases and exploited passwords by launching brute-force attacks. This type of attack used a tool that tried all possible combinations of the available keys on the keyboard. This usually takes time as there are a huge number of combinations to be tried out but is not reliant on the user's choice of password. Because of this, it has a good success rate if an attacker is determined to hack the system.

Users are often tempted to use easy to remember usernames and passwords which make their confidential information vulnerable to a security breach. Password guessing comes into use but requires some personal knowledge of the victim if it has to be effective. Initially, the attacker will gather personal information about the potential victim which can include pets name, parents name, birth date and phone number. After gathering this information, the attacker will try out various combinations of names and numbers. People tend to be very predictable and according to M.A Van der Linden (12), some of the most common password patterns are:

- Loved ones name + birth date/phone number
- Victims name + birthdate / phone number

Another way to exploit weak user passwords is through dictionary based attacks whereby a dictionary tool will keep on submitting the username and dictionary password guess until success is reached or the dictionary is exhausted.

Other vulnerabilities include poor authentication of users whereby users are only authenticated by their username and password only. This creates an opportunity for automatic spam created by non-humans, by that we mean computers. CAPTCHA's (Completely Automated Public Turing test to tell Computers and Humans Apart) [13] are currently used by a lot of Social networks during registration and when resetting a password to make sure that users are humans. They are sometimes described as reverse Turing tests [14] because they are administered by a machine and targeted at a human in contrast to the standard Turing test which is administered by a human targeted at a machine.

Chat programs and many other social networking sites indicate when a user has logged off a particular application/site. This can provide attackers with time to exploit previously found vulnerabilities when the user is out of sight, for example, a user logs off a social network, the attacker having maliciously obtained the users login details previously, might login using the victims credentials and post explicit material that might bring embarrassment to the user.

2.4 APPLICATION SECURITY RISK

Attackers can use many ways to infiltrate a system and according to the Open Web Application Security Project (OWASP), there are 10 critical security risks associated with web applications [4] which are listed below.

1. SQL Injection: This is an attack on databases in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. An attacker can do all sorts of things like deleting a database, stealing sensitive information such as user passwords and credit card information in e-commerce web applications.

2. Cross-site-Scripting (XSS): The attacker will try to execute scripts in a victim's browser which will hijack the user's session or send some sensitive information to the attacker.

3. Broken authentication and session management: The attacker will try to get the user's password using faults on application functions related to authentication and session management.

4. Insecure direct object references: The attacker will use the exposed reference of an object, such as directory, file, and database key to try and access unauthorized data.

5. Cross-site request forgery: The attacker will user the logged in victim's browser to send a forged HTTP request.

6. Security misconfiguration: The attacker will try to find a weakness in the security configuration and use it his advantage.

7. Insecure cryptographic storage: The attacker will try to get sensitive data stored as plaintext.

8. Failure to restrict URL access: The attacker will try to forge URLs to access hidden pages.

9. Insufficient transport layer protection: The attacker will try to get sensitive data being sent from one end to the other in plaintext. This can include password and other secret information.

10. Invalidated redirects and forwards: The attacker will try and redirect users to a phishing site to get user's logging information.

2.4 MODELLING THREATS

A.Shrivastave [15] suggested that in order to have secure web applications, it is important to analyse potential threats and model them. This approach can be applied using the threat modeling process, which is a process for optimizing application security by identifying threats, and defining countermeasures to prevent or mitigate them. Threat modeling helps shape web application design to meet security objectives such as confidentiality, privacy and integrity. Based on the decomposition of the web application, Threat Modeling evaluates the threats and risks to the application and chooses techniques to mitigate the threats. Security threats are modelled using attack trees which describe the decision making process an attacker would go through in order to exploit a vulnerability and compromise the web application.

Similarly, S.V.Casteele [16] presented a Masters thesis on threat modeling for web applications using the Microsoft STRIDE[6] model. The security objectives for the We Rock 24/7[7] web application were decomposed using Data Flow Diagrams (DFD).The STRIDE model was applied to the Microsoft We Rock 24/7 web application in order to categorise the threats. The DREAD model was used to rank threats according to their impact they had on the We Rock 24/7 web application .To summarise, the thesis highlighted the OWASP Top 10 most critical web application security threats [4].

Meanwhile, S.A.Klein [17] introduced a technique of applying threat modeling on a voting system. Voting systems have been demonstrated to be seriously insecure and vulnerable to malicious hacking. A threat model was developed that identified governmental power as an asset that required protection in voting machine security and that those attempting to compromise election integrity were likely to be highly motivated, technical competent and well financed. The potential pool of attackers included voting machine manufacturers and their suppliers, election administrators and political operatives.

D.De Cock et al [18] presented a paper on the analysis of threats that proliferated with the advent of smart cards being used in web applications. Their analysis was part of the "Designing Secure Applications" project (Microsoft Funded) of which its aim was to provide web application

[6] System developed by Microsoft for classifying security threats.
[7] Web application based on a virtual rock group that takes you through building an enterprise application using Microsoft technologies.

developers with a tool that allowed them to prevent the exploitation of a range of threats. One of these was the electronic identity card [18]. Griggs [19] constructed a comprehensive threat model for internet browsers by identifying well known threat models on the internet and then identified those which were more specific to browsers.

2.5 SUMMARY

The literature has provided us with insight to web application security risks. J.Scambray et al [7] outlined the motivation for why people attack web applications which aided in our knowledge on having a better understanding of the assets attackers are after. We then looked at social networking web applications and their weaknesses. The OWASP top ten [4] aided in our knowledge on the most common attacks experienced by web applications. We then looked at the literature on threat modeling in which provided techniques about on analysing and modeling potential threats.

The next chapter looks at the Elgg social networking engine that was chosen as a platform to create a social networking web application and applying the threat model approach to it in order to model its security.

CHAPTER 3

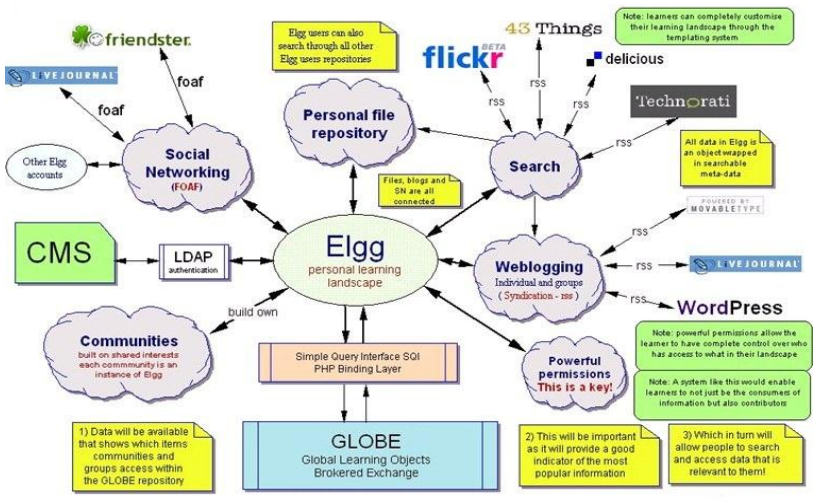
THE ELGG FRAMEWORK

Elgg was chosen for this project because of its robust framework that allows for development of all kinds of social environments from internal collaborative platforms for companies to university social networks. In this project, we will use the Elgg framework to build a social network that will serve as a test bed for threat modeling and testing vulnerabilities identified by the threat modeling tool.

3.1 INTRODUCTION TO THE ELGG SOCIAL NETWORKING ENGINE

Elgg is an open source social networking engine that provides a robust framework for building various online social environments such as online learning communities, private social networks for a university or business networking sites. Elgg is free to download and use as it is dual licenced under the General Public Licence (GNU) version 2, and runs on the Linux, Apache, MySQL, and PHP (LAMP) or Windows, Apache, MySQL, and PHP (WAMP) stack. Its creators, Tosh and Werdmuller term it as a "personal learning landscape" [20] since it incorporates elements of social networking, collaborative document authoring, e-portfolios (user's online identity) and web-blogging, see Figure 2. The Elgg application supports an autonomous learner-centred approach through web publishing and promotes the formation of learning communities in which communication, information and knowledge sharing can take place [21].

This promotes deep and significant engagement during the learning process through enhancing the e-portfolio. In this way, users can become contributors and recipients of knowledge, while establishing connections as well as building successful online learning communities.



Source: http://www.eradc.org/papers/Learning_landscape.pdf

Figure 2: Elgg personal learning landscape

3.1.1 e-Portfolios on Elgg

As illustrated in Figure 3, the profile page is a comprehensive e-portfolio that includes the user's online identity, profile picture, and links to blogs, bookmarks, files as well as user pages. The user has the option of deciding how much to reveal of their profile by applying different access control levels (public, logged in users and private). The same applies when groups are created, allowing only certain parts of the information to be made available to certain members of the group.

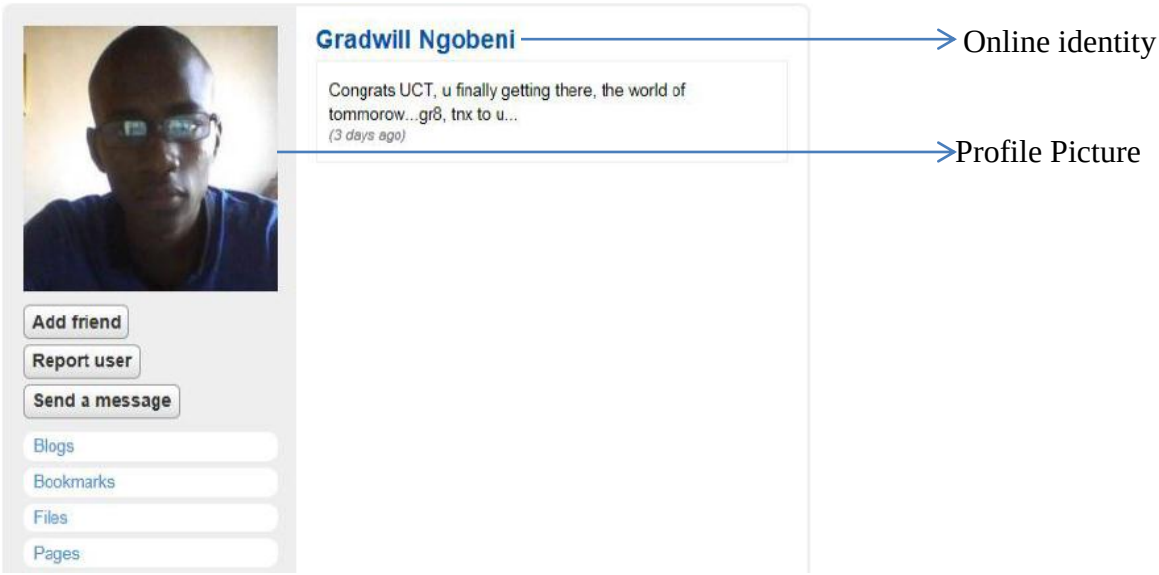


Figure 3: User e-Portfolio

3.1.2 Weblogging on Elgg

Elgg weblogs are frequently updated websites which are a simple form of personal publishing [22]. Weblogs give users an opportunity to write content that is unique to them, for example Individuals can write a weblog to share with the world their expertise on specific topics.

Posts or entries are published in a reverse-chronological order, meaning that the most recent posts appear first. Commenting is an important part of weblogging that takes a single post and turns it into a conversation that other users can engage in. For example, a user could post a comment about an upcoming soccer match and engage other users to post comments or opinions as shown on Figure 4. Elgg also expands this blogging model by providing access control and cross tagging capabilities that a user can assign permissions to other users wanting to view his/her blog posts and/or search for other related blog posts.

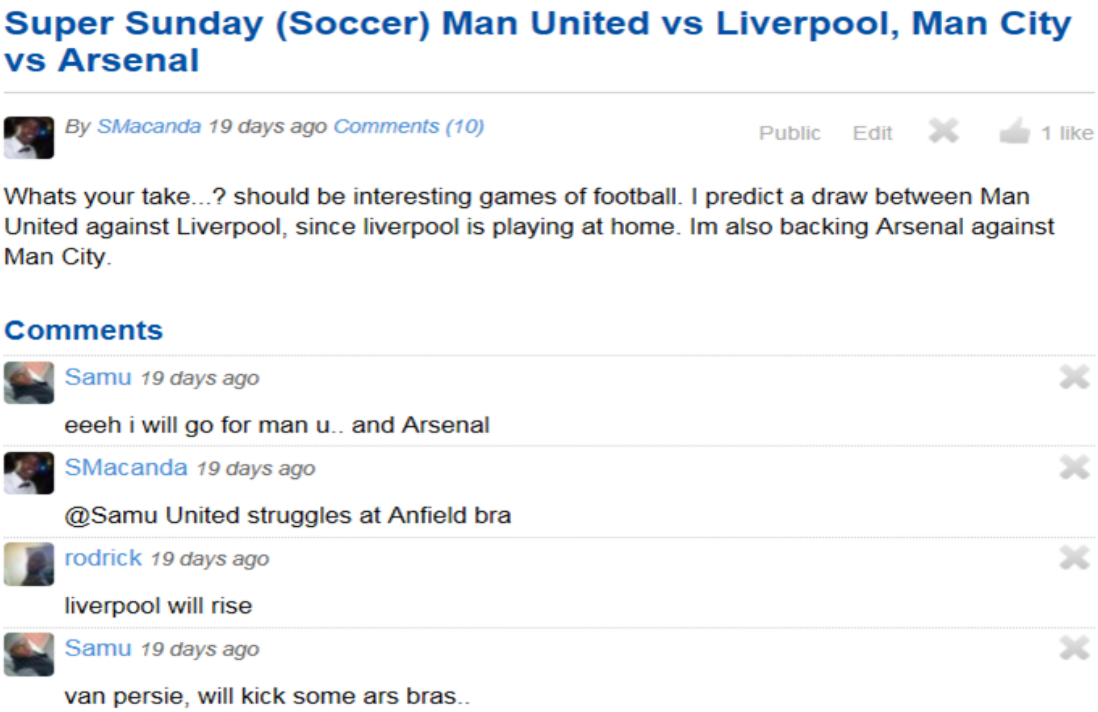


Figure 4: Weblog on a Soccer Match

Blogs are quite popular; because statistics from February of 2011 indicate that there were approximately 156 million public blogs present [23]. Furthermore, a survey conducted in July of 2012, there were approximately 70 million Wordpress and 39 million Tumblr blogs in existence [24].

3.1.3 Social Networking on Elgg

Elgg offers two social networking features, namely, keyword tags for content and friends or groups for users [21]. The keywords enable users to find other users who have published on similar topics. These keywords appear as links at the bottom of a post that a user can click in order to visit a webpage containing posts with the keyword. Elgg also offers automatic tag suggestion for further browsing of content and other users, this saves users the work of manually tagging content and also aids in eliminating errors that can be introduced when attempting to do manual tagging.

Like Facebook, Elgg allows users to invite other users and categorize them as “friends’ by adding these users to a pre-defined friends catalogue. Friends can view each others profiles, comment on each others status updates and share resources such as photos, videos and

documents. New users can find friends by clicking the potential friend's username link on blog posts or wire posts (similar to Facebook's status update); this will redirect them to the users profile page where they can invite them as a friend. The invited user has the option of accepting or declining the invite, if they accept the invite, both users become "friends" on the social network and their friendship status is posted on the main activity page. In this way, other users who are friends with each of the users can see the newly established relationship and may decide to invite the other party if they know them or interested in being their friend, see Figure 5.

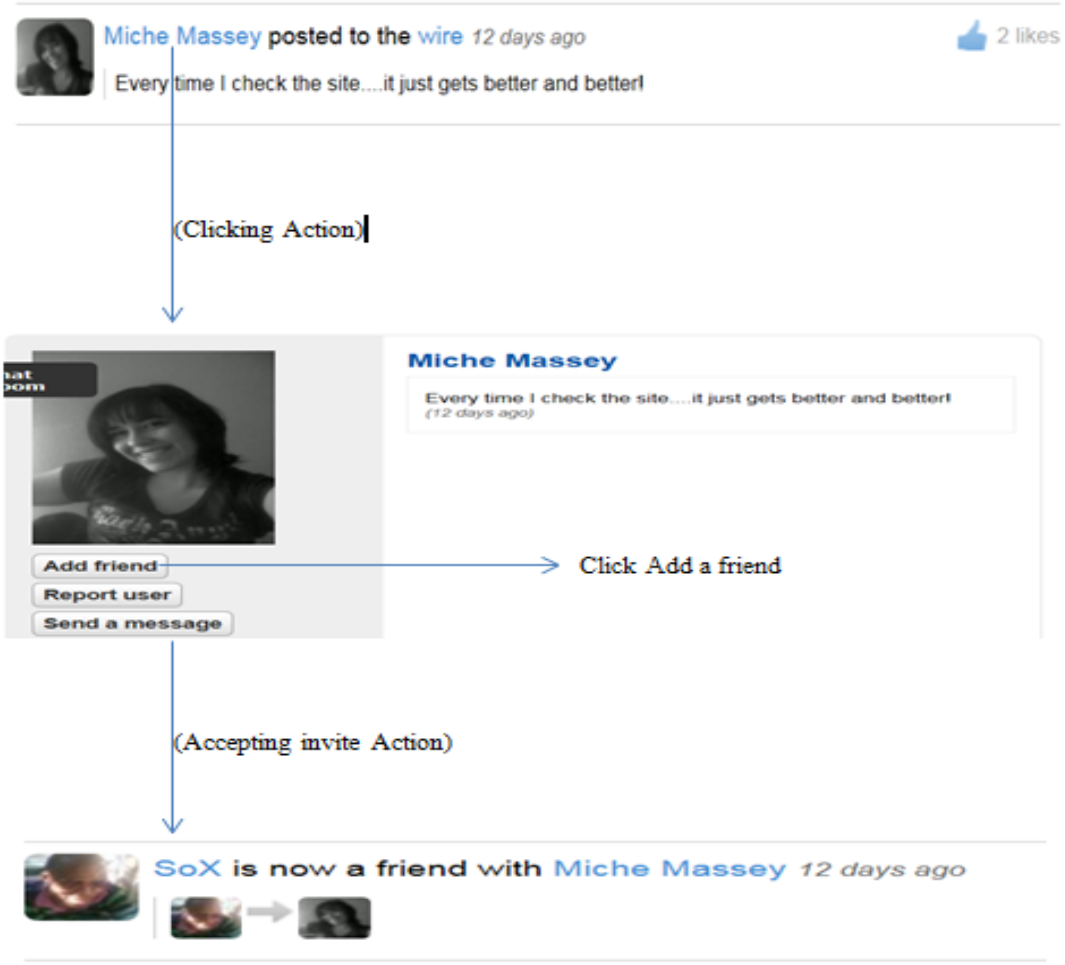


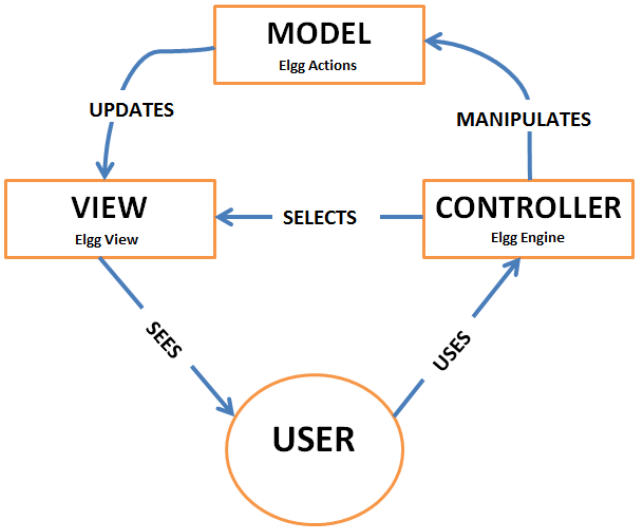
Figure 5 : Inviting a Friend on Elgg

Another aspect of social networking is that users can create groups with the objective of sharing photos, videos and posts. Elgg groups have access level features which allow a user to specify who is able to join the group and who is able to view certain information. Groups can be public meaning that anyone can join a group or they can be private meaning that the creator of the group can invite other users to be part of the group.

3.2 ELGG ARCHITECTURE

Elgg is built on the Model-View-Controller (MVC) architecture which is basically a design pattern for user-facing[8] software that isolates the representation of information from the user's interaction with it [25]. The MVC pattern was conceived in the late 1970's by Trygve Reenskaug at Xerox PARC as part of the Smalltalk programming language and has widely been adapted as architecture for web applications.

The model divides the Elgg application into three parts which are Models, Views and Controls. Apart from dividing the application into three components, The MVC architecture also defines the relationship between them. Looking at Figure 6, a user will interact with the controller which is basically the Elgg engine, the controller in turn will manipulate a model (Elgg actions) and select view (User interface) that a user can see. The MVC's components will be discussed in more detail in section 3.2.1, 3.2.2 and 3.2.4.



Source: en.wikipedia.org/wiki/Model-view-controller by Regis Frey, modified by Molulaqhooa Maoyi

Figure 6 : MVC Component Collaboration

In the next subsections, The MVC architecture will be discussed in more detail with respect to the Elgg Social Networking Engine. Concepts such as Model, View and Controller will be explained in more detail.

[8] Software the a has a Graphical User Interface that a user can interact with

3.2.1 Models

Models are application components that implement the logic for the application data. Models often retrieve and store model states in a database [26]. An example is a Blog object retrieving information from a database, make operations on it and then write updated information back to the Blogs table in the database.

Elggs stores Models in the /actions directory which contains the core actions of the system such as logging in, creating, updating or deleting content. Actions are executed when users post forms such as submitting a blog comment. The "action code" adjures the submitted data and makes the relevant modifications to the database.

3.2.2 Views

Views are components that useful in displaying the application's graphical user interface. These views are usually created from the model data. An example would be an edit view of the blog table which displays the graphical representation of the blog with comments.

Views in Elgg are stored in the /views directory and each directory under it contains a view type. Views are responsible for creating output from layout of pages, footers down to form inputs. They can also be used to develop mobile interfaces and allow for advanced features like automatic RSS[9] generation [27].

3.2.3 Controllers

Controllers handle user interactions; interact with the model and select views to display as seen on Figure 6. For example, the controller might handle query-string data and pass it to the model which in turn will use this data to query the database.

The Elgg engine corresponds to an MVC controller. The core classes of Elgg together with database access code and session handling are found in the /engine/lib folder.

[9] RSS (Really Simple Syndication) is a family of web feed formats used to publish frequent updated infomatiom

3.3 PLUGINS

Elgg is an extensible platform in building custom social networking sites through the development and use of plugins. A plugin can be written to extend or replace just about any portion of the Elgg core functionality, for example, an email validation plugin can be coded and then overrides the Elggs core email validation functionality (see Figure 7). A plugin can be used to add a new feature to Elgg, for example, creating a blog plugin adds the functionality of blogs on Elgg (see Figure 8).

Source: http://www.slideshare.net/intunex/how-to-build-your-own-social-network-with-elgg-4987242

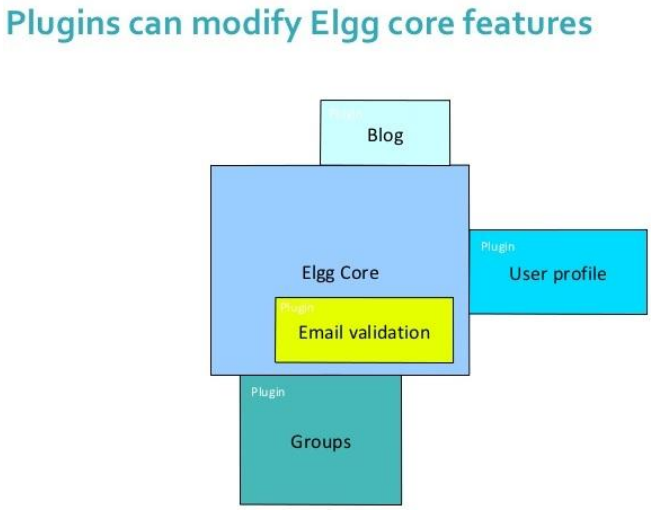


Figure 7 : Plugin modifying Elgg core

Figure 8 : Plugin adding new feature to Elgg

Plugins can modify other plugins, for example, a plugin for groups can be modified by another plugin called group members. See Figure 9. This gives it more added functionality.

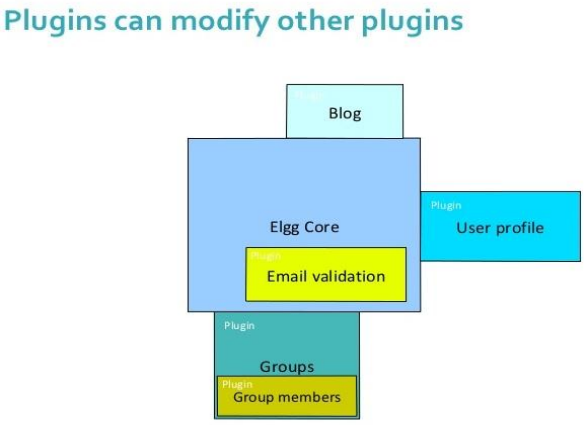
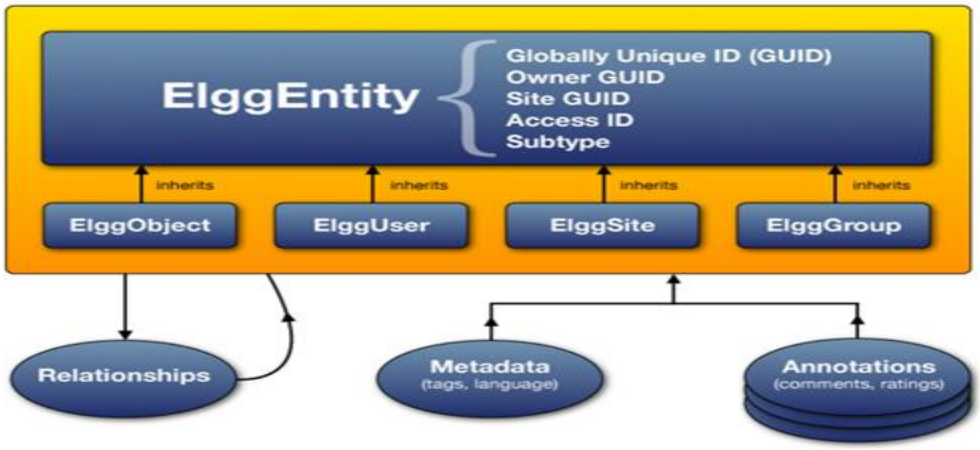


Figure 9 : Plugin Modifying another Plugin

Each Elgg plugin resides in its own subdirectory which is located under the /mod folder. Every plugin should have a start.php and a manifest.xml file in order to be recognized by Elgg. The start.php initializes the plugin while the manifest.xml file stores basic information about the plugin.

3.4 DATA MODEL

Elgg is built on atomic data units called entities. A user is an entity, a blog post is an entity and a group is an entity. There is a base class in Elgg called ElggEntity. All other classes extend that class, for example, the ElggUser class extends ElggEntity class to provide user functionality. Besides the entity class, there are three helper classes that make it simple to add new functionality which are the ElggRelationship, ElggMetadata and ElggAnnotation. The ElggRelationship class allows connections between entities i.e. a user can "friend" another user. A user can join a group or tagged in a photo. All of this can be done through the ElggRelationship class. When a photo is tagged, the tag is stored as an instance of the ElggMetadata class and when a user comments on a blog, which is stored as an ElggAnnotation. Figure 10 belows shows the Elgg data model.



Source: http://docs.elgg.org/wiki/File:Elgg_data_model.png

Figure 10 : Elgg Data Model

3.4 SUMMARY

This chapter served as a means to describe the power of the Elgg social networking engine. Since it is a framework, it requires you to put some effort into developing various components in order to get your social network the way you want it to be. On the other hand, it is very extensible and easy to add functionality you want through plugins.

CHAPTER 4

HACKMI2 SOCIAL NETWORK

4.1 INTRODUCTION

As discussed in Chapter 1, the aim of this project is to build a social network that is powered by Elgg's social networking functionality. This social network, Hackmi2, will serve as a test bed for the purpose of studying the threat modelling approach on analysing the security of social networks and for comparing the three threat modeling tools (Microsoft TAM, Microsoft SDL and SensePost CTM). In this section we look at the overall Architecture of the Hackmi2 social network, the underlying technologies and the different components that make up the social network.

4.2 TECHNICAL SPECIFICATIONS

4.2.1 Hardware Specifications

Table 1 below lists the hardware specifications for the Social Network.

Name	Proline officeware Desktop machine
Processor	Intel Core i7 -2600 CPU 3.40 Ghz
Random Access Memory	16 GB
Hard drive space	3TB

Table 1 : Table of Hardware specifications

The reason this hardware was chosen was that Social Networks tend to have a huge database of users who create profiles, chat, and upload photos, videos and post blogs. These actions tend to be processor intensive as many users might be accessing the social networking site concurrently which can impact negatively on the loading time of the site.

4.2.2 Software Specifications

Hackmi2 utilizes open source technologies that can be divided into 4 core technologies: Apache, PHP, MySQL Server and Ubuntu Server Edition. In the following page, we will give a description of each of these technologies in some more detail.

Ubuntu Server: This is the Debian Linux distribution operating system that is used on servers. It does not come with a graphical user interface and user interaction is only text based. The current version running on the server is the 12.04 LTS precise edition which is the latest released version of the operating system.

MySQL Server: This is the most used and popular open source Relational Database Management System (RDBMS) with the latest version being the 5.5.24.

PHP: This is a recursive acronym for Hypertext Preprocessor [28] is an open source scripting language that is used for web programming and can be inserted into HTML.

Apache: The world's most popular webserver because its open source and supports most web application. Its function is to deliver Social Networks web content that can be accessed via the internet; this includes stored files and webpages.

4.3 THREE-TIERED ARCHITECTURE

Hackmi2 utilizes a three-tiered architecture to segment the application into logical layers, namely the client, webserver and data layer. This section begins by with defining the different layers and how they interact with each other. We wrap up this section with a high level overview of the three-tiered architecture shown in Figure 11.

i) **Client layer**: This layer also known as the presentation layer presents data to the user and allows for input and data manipulation. On the Hackmi2 social network, the presentation is the Social network's front-end which can be accessed using a browser as depicted in Figure 4. Users are able to post pictures and post blogs using the presentation layer and are able to interact with the social network.

ii) **Webserver:** This is also referred to as the middle-tier. This is where the business logic of the Social network takes place. This is where the Elgg Social Networking Engine is installed and in addition requires Apache 2 and PHP 5.

iii) **Data layer:** This consists of the database management system which is MySQL and data folders to store files such as photos and audio.

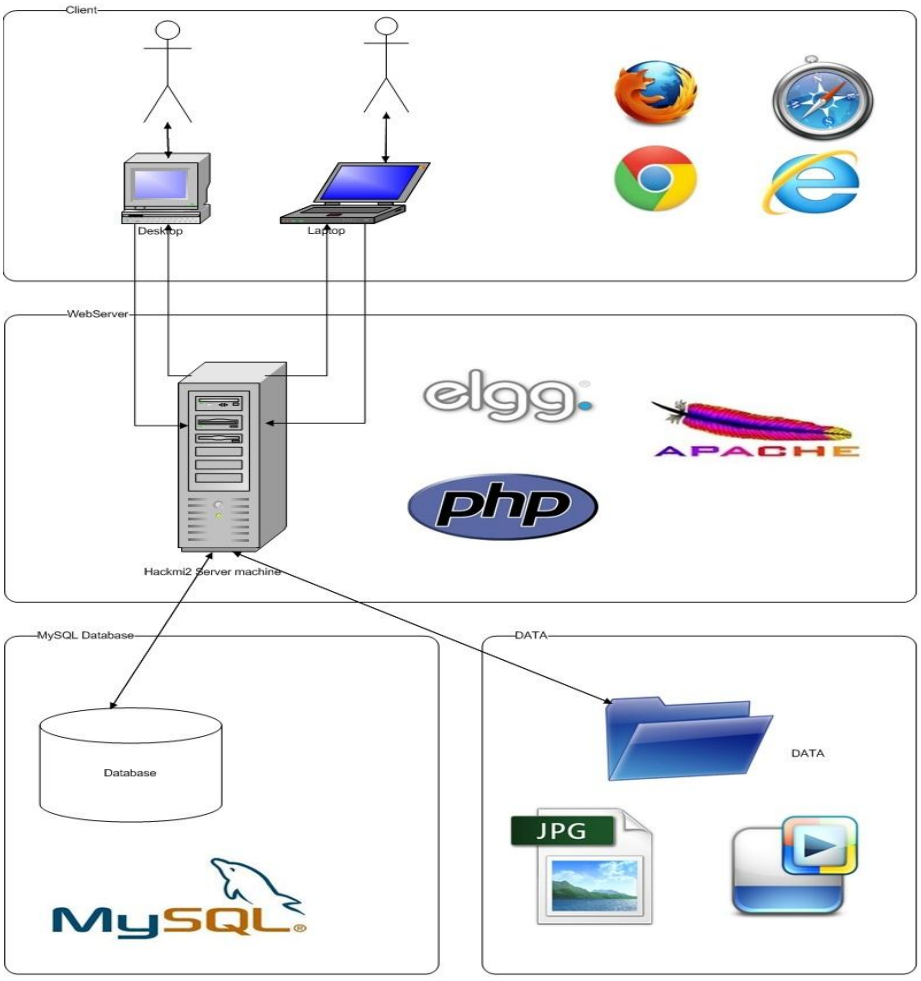


Figure 11 : Hackmi2 Three-tiered architecture

The three-tiered approach provides benefits such as reusability, flexibility, manageability and scalability of applications [29]. Components can be shared and reused and distribute across networks of computers as needed. Large and complex projects such as social networks can be divided into simpler projects and assigned to different programming teams

In the next section we will look at features implemented on the hackmi2 Social network

4.4 IMPLEMENTED FEATURES

The Hackmi2 Social network incorporates well known social networking features that are found on other Social networking sites such as Facebook, MySpace and Twitter. According to D.M.Boyd and N.B.Ellison [30], social networking sites share a number of technical features that allow users to: construct a public/semi-public profile, display a list of other users that they share a connection with, and view their list of connections.

The features that were incorporated into the Social network included:

- Registration of new users
- Retrieval of Lost passwords
- Updating Profile
- Uploading Pictures and File
- Updating Status
- Blogging
- Commenting of blogs and status updates
- Bookmarks to bookmark interesting pages
- Uploading of Audio files and Playing them on the site
- Chat room
- View of Horoscopes
- Creation of Groups
- Inviting Friends and "Unfriending".
- Creating Pages similar to Facebook pages for companies and celebrities.
- Viewing Friends profile
- Searching Function
- Sending Messages
- Change Passwords, display name and email
- Reporting of users

A. Index Page

Before using the social network the user will first be greeted by the Home screen, see Figure 12. To the left of the home screen, users can read the Privacy and Terms and conditions of use for the Social network (Appendix A.1 and A.2). Registered users can login using their valid username/email and password credentials.

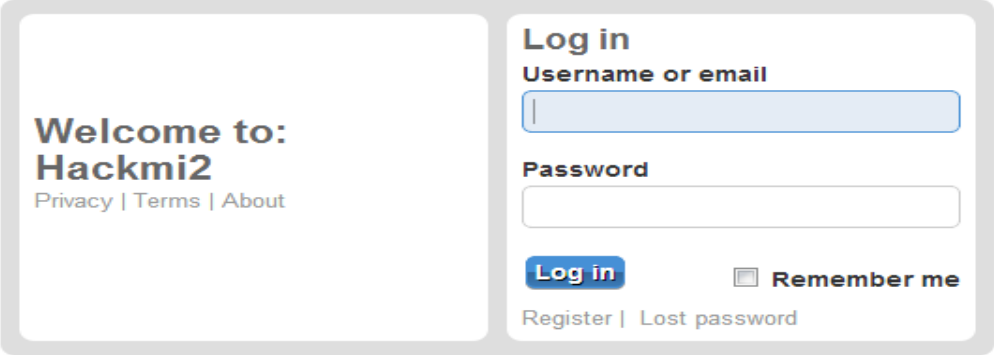


Figure 12 : Home Screen

B. Lost Passwords

If a registered user has "lost" or forgotten their password, they can easily retrieve it by clicking on the Lost password link (see Figure 12) and they will be redirected to a page where they can enter their username or email address they used during registration and a link to reset their password will be sent to their email address, after resetting, a new computer generated password will be sent to the users email address. See Appendix B.

C. Register

Unregistered users have to first register their details before they can access the social network. This takes form of a registration form where unregistered users must fill out their display name, email address, username and Password. In addition, users have to verify they are human by selecting the correct answer from a CAPTCHA. This is to prevent spambots[10] from carrying out automated registrations which will fill up the social networking with junk information. After registering successfully, an email will be sent to the users email address and it will require the user to click on a link in order to verify the account (see Appendix C).

[10] Automated computer programs designed to send spam.

D. Activity page

The activity page is the first page that appears when users login, see Figure 13. It displays the users profile (A) and a welcome message. It shows the date the user first joined the social network and their last login date and time (B). A user is also able to view their daily horoscope and by clicking on "your day today" (C), this is to provide extra features that will keep users interested in the social network. The user is able to post status updates using the form under "Share your thoughts with everyone" (D). The user is able to view other user's activity on the social networks, which are their status updates, blog posts and photos uploaded through the activity pages. This could be adjusted to view only personal activity or friend's activity (E). A user is able to logout of the social network easily by clicking logout on the top right hand corner, see (F). Users can use the search box (G) to easily search for users and content in the social network, for example if a user reads an interesting blog and forgets to bookmark the page, provided they know the name of the blog, they can search the site and the social network will return the results, see Figure 13. A user is also able to view Latest Bookmarks and latest files uploaded files (G) onto the social network.

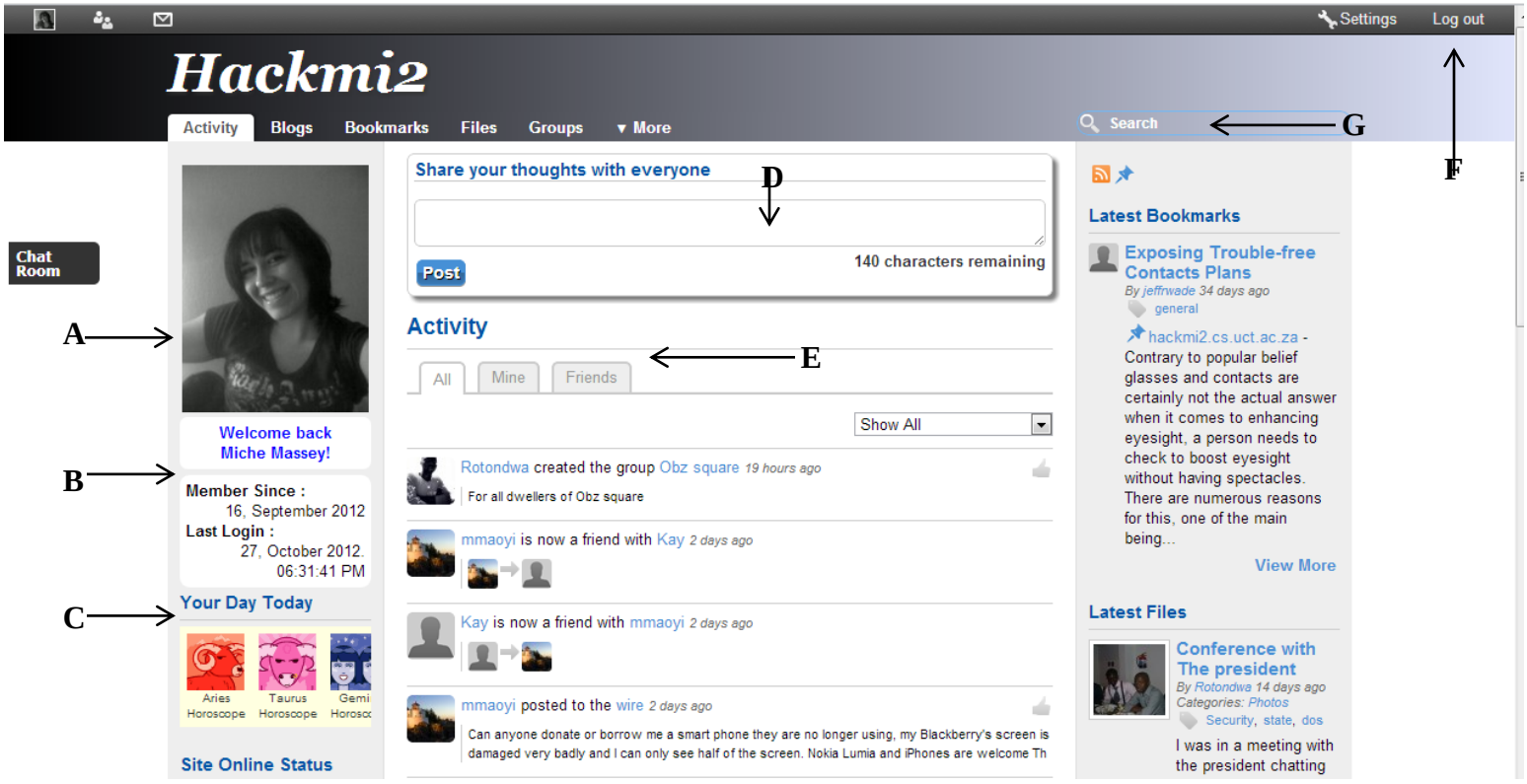


Figure 13: Activity Page

E. Updating Profile

A user can update their profile picture by clicking on the Edit Avatar as seen on Figure 14 and this will take the user to the page where the user can upload a new avatar or edit it. A user can edit their profile by selecting “Edit profile” and they are able edit their profile fields including selecting who is able to view certain information. For example a user can select the date of birth profile field to be private meaning only the owner of the profile can view their birthdate whilst other information like Interest and languages are public and can be viewed by everyone.

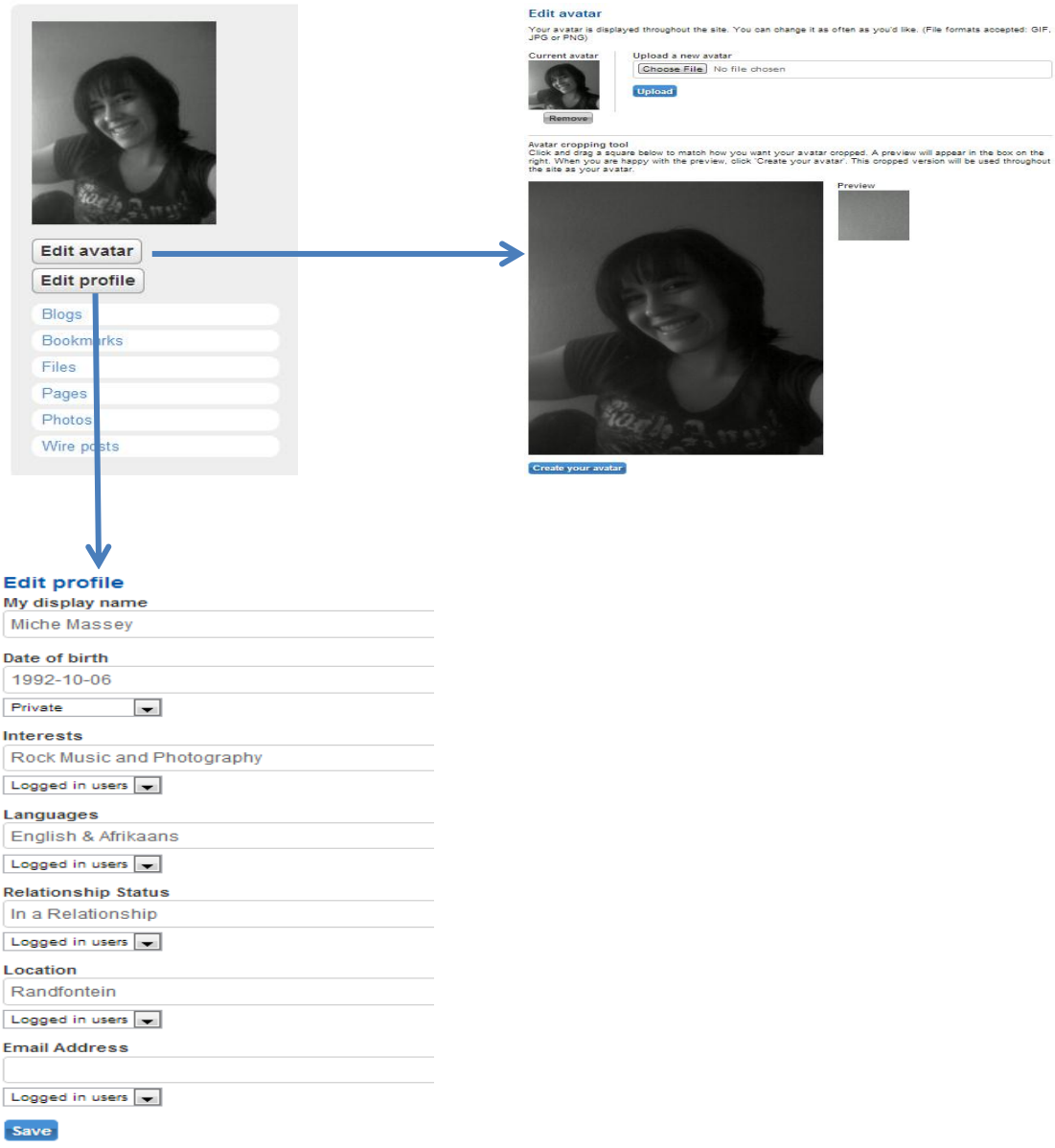


Figure 14 : Updating Profile

F. Blogging

A user has the option to view all other user's blog posts, their own personal blogs or their friend's (A), see Figure 15. Users are also able to create their own blog posts by clicking on "Add blog post" (B) and have an option of whether people can comment on their blog posts the and who gets to view their blogs (C). This is to give users an option on what information they would like to share to the public, their friends or just keeping it private.

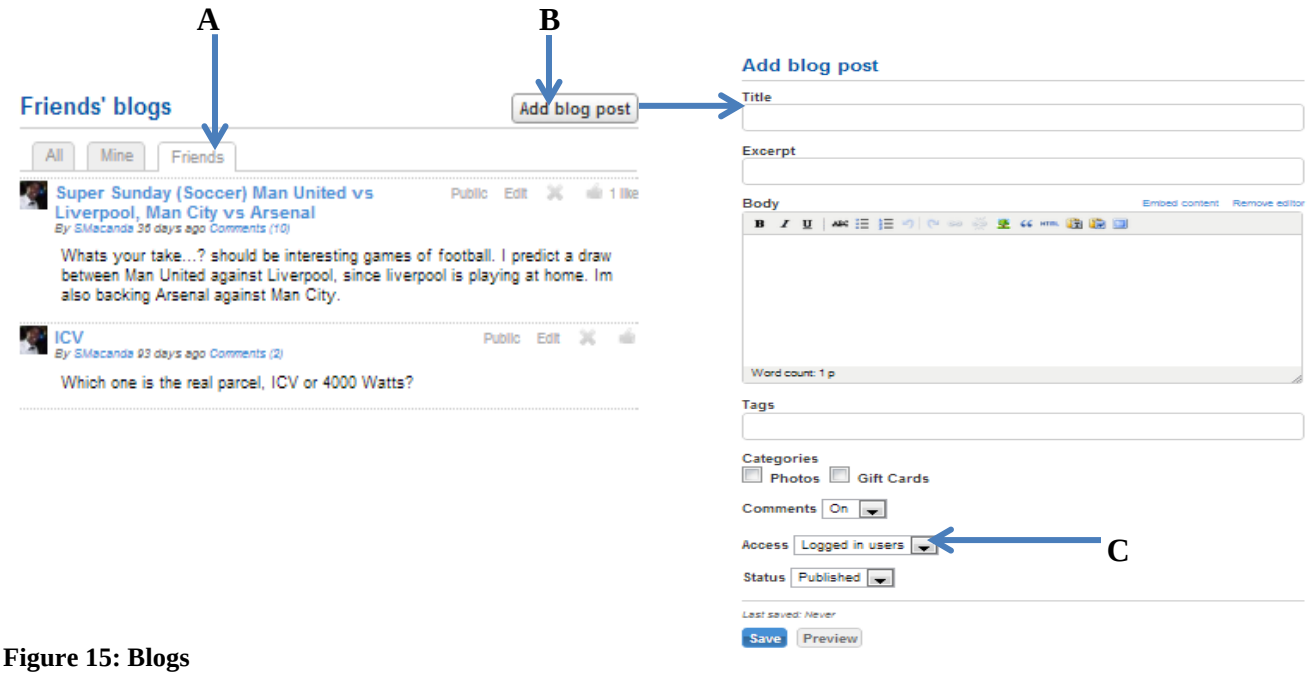


Figure 15: Blogs

G. Multimedia

Users are able to upload music and videos and play them on the social network and other users are able to view or listen to the videos and music a user has uploaded without having to download the song and playing it on their computers, see Figure 16. Users are also able to write comments about the music or video and rate it using a "Like". This is similar to the concept of YouTube where users are able to upload videos and music and users are able to "like" them and write comments about them.

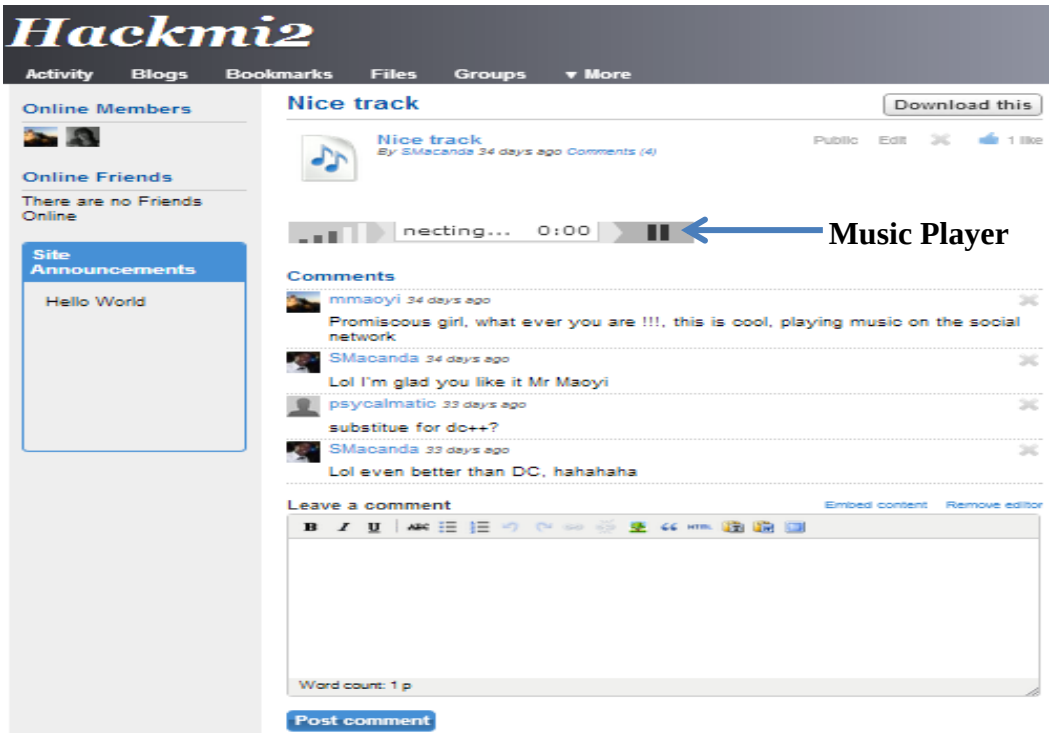


Figure 16 : Playing Multimedia on the Social Network

Users can also upload photos and create albums (see Figure 17) and view them as a slide show as shown in Figure 18.

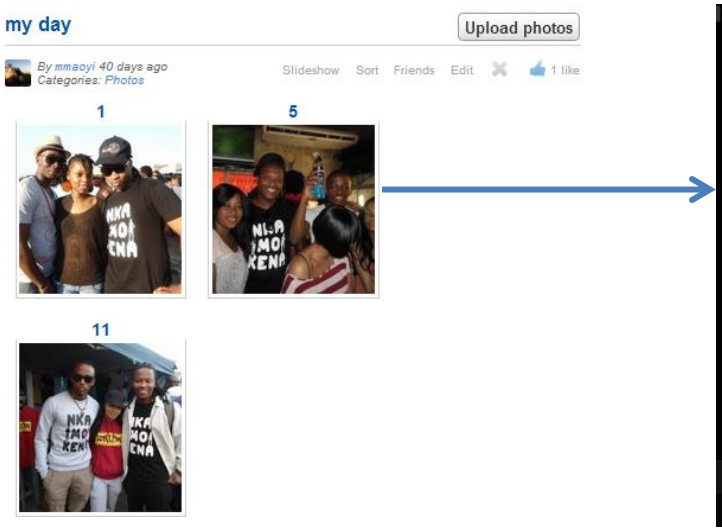


Figure 17: Photo Album

Figure 18: Slide Show of Photos

H. Chatting

Hackmi2 supports private chatting between two users and also group chats (see Figure 19 and 20). The name on the blue bar denotes a friend's user name in order to keep track of who a user is chatting to. Figure 19 shows two users chatting, masseym and mxymol001, and masseym is asking myxmol001 what he is doing and myxmol001 is writing a reply about being busy with a thesis most probably due soon.

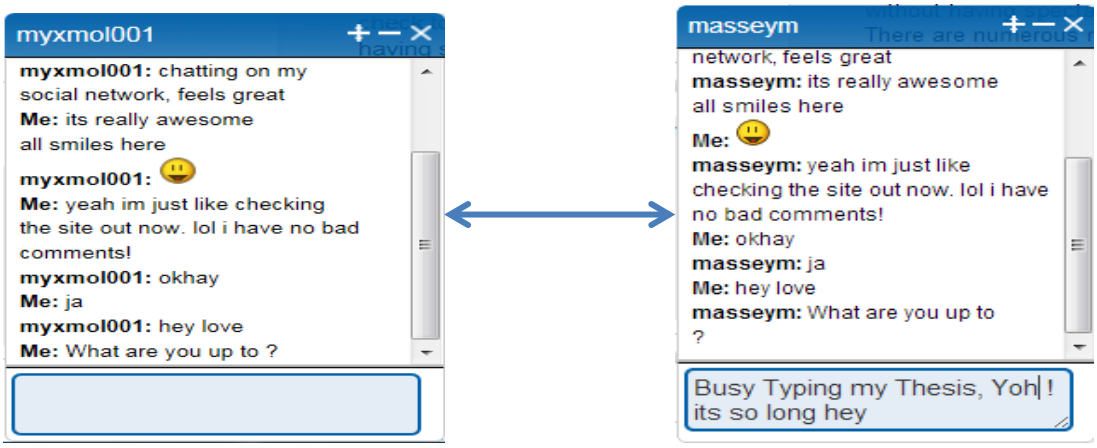


Figure 19: Chatting between two users

Figure 20 shows that the social network consists of six chat rooms which are Supreme talk which is aimed at topics which are interesting such as great discoveries in science such as Supernovas, Dark Matter and anything nerdy; Boring talk (most probably public policy and legislature); Fun talk where users can share jokes and funny stories; General talk is self explanatory; UCT-Crew is a chat room aimed at UCT students to chat about campus life and meet fellow UCT students; Innovative talk is aimed for the innovators and those who would like to make change in the world through innovative ideas.

Figure 20: Group Chatting

I. Creating groups

Users can create groups and other users can join them. Groups can be public, meaning anyone can join, and they could also be private, which means that users join by an invite basis only. Users invited to a private group may accept or decline to join the group. Users also have an option to leave a group and invite friends to the group if they are already members of that particular group (see Figure 21). Groups can have their own blog post, bookmarks, files discussions and pages. Figure 21 shows the Smuts hall group which is not a really active group and has only 2 members but this is just to show the powerful features of Groups on the hackmi2 Social network.

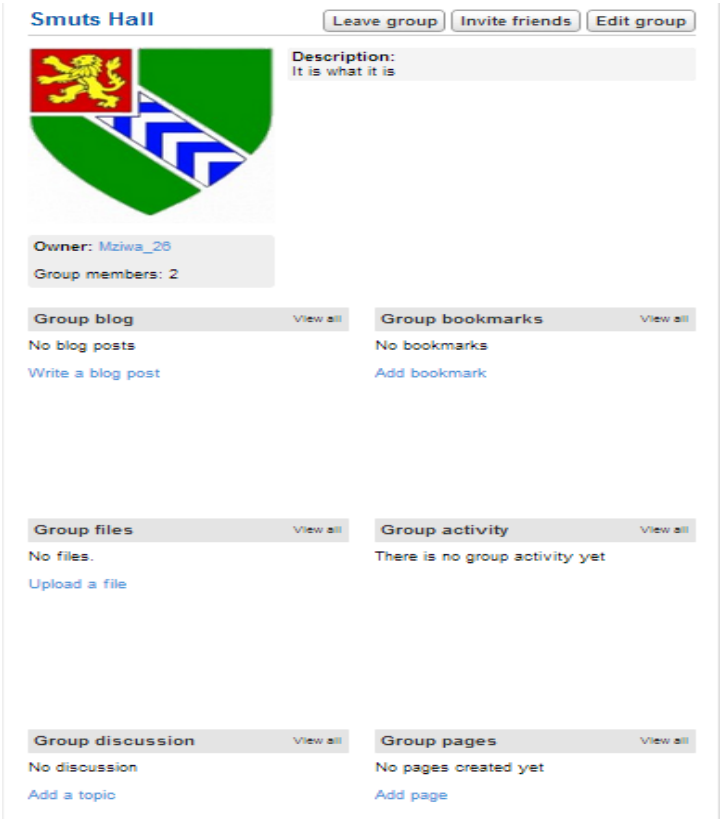


Figure 21: Smuts Hall Group

J. Messages

Users are able to send messages similarly in the same way one would send an email by selecting the recipient and writing the subject. After that a user can write their message using the Tiny MCE editor which was incorporated into the social network for editing content such as blog posts, comments and messages as seen on Figure 22.

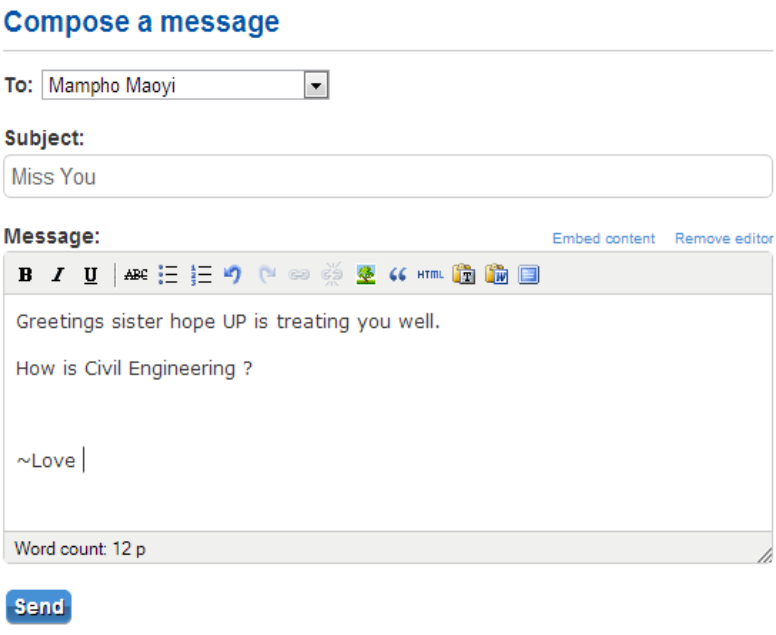


Figure 22: Sending a Message

CHAPTER 5

THREAT MODELLING

5.1 INTRODUCTION

This section is centred on the threat modeling approach to identify the threats and vulnerabilities in a system. Most systems contain some form of valuable asset that needs to be protected from unauthorized access. For instance, in a social network, it makes sense to protect the password database from being accessed by users in general, and malicious users in particular. Threat modeling is useful therefore in detecting the sorts of vulnerabilities in the system that might lead to valuable assets being compromised. As mentioned before, we consider the security vulnerabilities from the attackers view in order to understand how to design better security mechanisms and countermeasures.

5.2 THREAT MODELLING TERMS

Assets: Assets are resources that need to be protected from attackers and these include:

- **Physical Assets** such as the server computer hosting the Hackmi2 Social Network.
- **Abstract Assets** such as Social Network reputation, availability of resources and service.
- **Electronic Assets** such as databases, and data on the server.
- **Transitive assets** are assets that act as a gateway to other assets [12].

Attack Tree: A sequence of operations that must be executed to identify a threat.

Entry Points: According to M.A. Van der Linden [12], Entry points are the locations at which the "control or data crosses the boundary of a system is being modelled" and that can potentially be exploited to provoke an attack. For example, in a house, the most obvious entry points would be doors and windows while the not so obvious might be air-conditioning ducts and drop ceiling panels. Therefore, all the access points of data into the Social Network must be considered, as well as interactions with the other applications, network data, and administrative controls.

Exit Points: These are points where data leaves the system which can be in the form of log files. Exit points are like the rubbish bin outside the house.

Threat: A threat is what an attacker might attempt to do to compromise a system. These threats are then listed and compiled into a threat profile.

Trust levels: These are set of rights given to any entity based on what is known or thought to be known about that entity.

5.3 WHAT IS THREAT MODELLING

A threat model is "A systematic, non-provable, internally consistent method of modelling a system, enumerating risks against it, and prioritising them." [31]. As this is an attack focused risk assessment [31], threat modelling is usually implemented during the design phase of the Software Development Life Cycle (SDLC). In order to identify the methods an attacker might use to exploit vulnerabilities in a system. Threat modelling defines the security of the application which helps to scope and set boundaries and constraints for the system. Decision making use of scenario modelling can be employed by a threat model whereby key issues that represent the risks such as authentication and authorization are highlighted since the scenario modelling is a skeletal overview of the before and after picture.

The benefits associated with such a model are the ability to identify and investigate potential threats that could be mitigated early. This helps to prioritise threats and identify high impact vulnerabilities. In addition, threat modelling exposes the logical or architectural vulnerabilities in a system, which is useful in to validating the security design of the system before the development process initiates.

5.4 MODELLING APPROACHES

There are three main approaches to threat modelling and depending on who is evaluating the system. According to N. Sportsman [2], one could look at the system in any of the following ways:

- **Asset Centric**

  Evaluating the system from asset classification such as personal information.

- **Defence centric**

  Evaluates weakness in security controls and looks for attacks against each element of the model.

- **Attack Centric**

  This where the system is evaluated from the point of view of an attacker and how they will go about exploiting the system and what they could possibly try to attack. For the purpose of this project report, we will be studying the attack centric model in depth.

  In order to successfully portray an attacker, one has to have a good idea of who the attackers are and what their skills and motivations are. This helps to write correct and meaningful security test cases. It requires one to look outside their box of "customers" (users in the social network) and look at the bigger pool of "consumers". According to M.A. van der Linden [12], there are three main parts to this formula: what, why and who. Because you already know who your customers are, the attack centric model is a way to become aware of the potential attackers your security test case is trying to prevent.

  What?

  The most basic question is what exactly attackers aim for. The answer to this question is basically the social networks assets that might be at risk. In threat modeling, this list of assets and the one in the threat modeling tool should be in agreement, meaning they have to be the same. If they are not in agreement, their differences should be analysed and resolved.

Below is a list of potential assets that an attacker might look at:

User Assets

- User Login information
- User permission (e.g. Administrator rights)
- User data

System Assets

- Availability of the Social Network and its services

- Data and information of the Social network

Network Assets

- Availability of Network (e.g. UCT network)
- Data and information stored on other machines on the network
- Network information

Why?

Why would someone want to attack an asset? If there was no motivation to carry out an attack then there would no point in doing so. M.A. van der Linden (12) suggests the following reasons as motivations for attacking assets:

Cyber-Terrorism

- Cyber-extortion
- Defamation

Personal and Group Recognition

- Cyber-Tagging
- Headlines in the press

Data

- Data theft
- Data destruction
- Data modification

Resources

- System resource theft
- Network resource theft

Who?

Who are the people carrying out attacks? This is a tricky question as there can be any number from both sides of the fence, for example, malicious attackers and security researchers of all skill levels. The list below lists the different types of potential attackers.

- Security Researches
- Script Kiddies
- Individual Hackers
- Organized crime
- Angry current or Former employees
- Competitors

5.5 THE THREAT MODELLING PROCESS

Threat modelling is an iterative process because it is highly improbable that one can identify all possible vulnerabilities in a single pass. According to J.D. Meier et al [32] applications rarely stay the same forever and so, as an application evolves, the threat modelling process should be repeated to account for that change. Below is a diagram that outlines the threat modeling process and its various components.

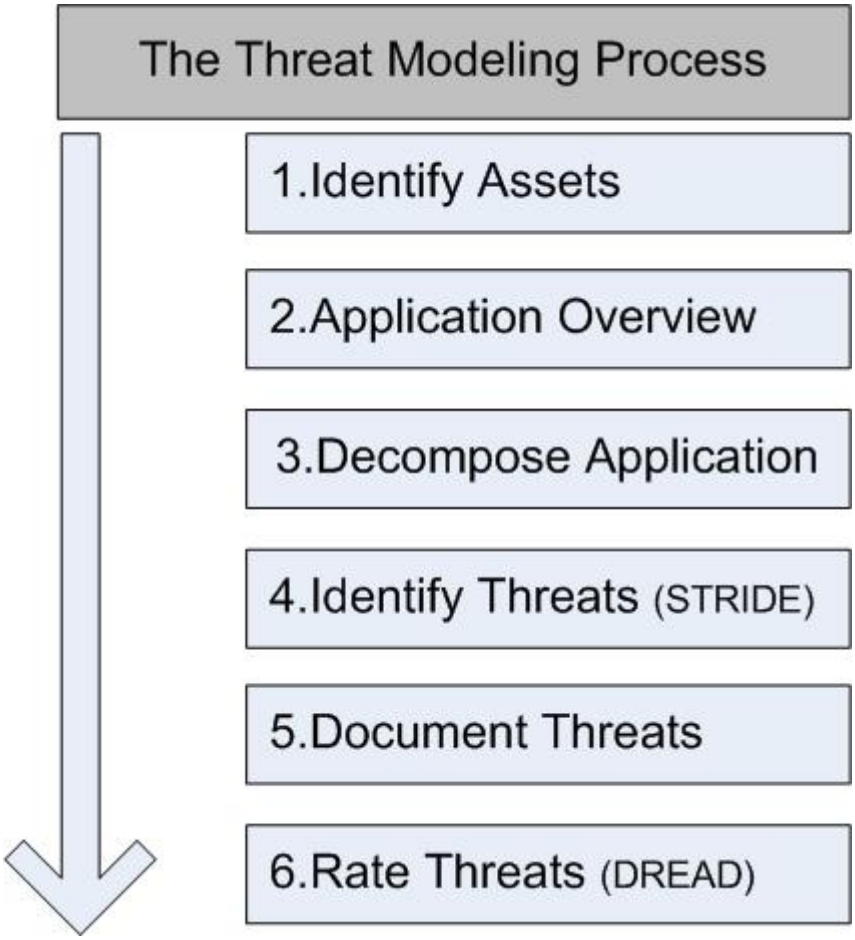


Figure 23 : Threat modelling Process

5.5.1 Identify Assets

With reference to Figure 23 of the threat modelling process, from an attacker's point of view the first step would be to identify the assets that need to be exploited, this could be confidential data such as names, email addresses on user databases. According to J.Scambray et al [7], identifying assets of value makes a difference between an extremely useful threat model and a simply mediocre one. Setting succinct objectives helps clarify what the priorities are and what are not, what are the musts versus coulds and shoulds.

5.5.2 Application overview

Figure 23's second point, here we look at what the system does and how it uses or accesses resources. This is an important step because having an overview of the application allows the threat modeling tool analyse how certain resources can be used and misused by documenting use cases. Use cases help to put the systems functionalities in context for easier understanding.

Here are the sample use cases for a Social Networking application:

- User Logs into the application
- User updates personal profile
- Administrator views user personal profile

In the above cases, if vulnerabilities exist with user profiles, the business rules may be misused for example a user who is not an administrator modifies another user's profile.

5.5.3 Decompose the application

Looking at Figure 23 point three, the attacker decomposes the application to create an attack profile for the system based on known vulnerabilities. There are also certain steps that an attacker might want to take such as identifying data flow between the social network and the server using Data Flow Diagrams (DFD's). DFD's are useful in decomposing the system by providing detailed representation of the system components complete with boundaries and connections to other systems. In addition, it is also vital to identify entry points in the system that might serve as entry points for any attack. Entry points are locations where data enters or exists the application; these may include authentication forms, web applications listening for HTTP request, profile-related web pages and searching.

5.5.4 Identifying threats

Looking at Figure 23 point four of the threat modelling process, the attacker identifies the threats associated with the system, this just provides a laundry list of those threats that might or might not exist in the system and so the next step in identifying threats is to classify them into categories using the Microsoft STRIDE Model which will be discussed in section 5.6. This means that threats are grouped together according to one or more criteria to make it easier to understand the threat.

5.5.5 Document threats

After the threats have been identified the next step would be to document them using a template that consists of several threat attributes that include the threat description, threat target, attack technique and mitigation strategy or countermeasure. Table 2 shows an example of a threat document [32].

Threat Description	Attacker obtains authentication credentials by monitoring the network
Threat Target	Web application user authentication process
Attack Technique	Use of network monitoring software
Countermeasures	Use SSL to provide encrypted channel

Table 2: Threat Document

5.5.6 Rate Threats

In the final stage of the threat modeling, the documented threats are then rated based on the risk they pose. This allows for addressing the threats that possess the most risk first and then continue to resolve other threats. Sometimes it may not be viable to address all the threats as some have a very small chance of occurring and their damage if they should occur will be very small.

The next section will be looking at the Microsoft STRIDE model in detail and how it classifies and categorizes threats.

5.6 MICROSOFT STRIDE MODEL

The Microsoft STRIDE model is a system for categorizing the discovered vulnerabilities that have been identified. Classifying vulnerabilities by categories makes the job easier in understanding what the attacker is capable of and can also aid in prioritizing vulnerabilities. Because vulnerabilities may belong to more than one category, the STRIDE model will classify the vulnerability according to their root cause, for example, if a vulnerability is associated with data tampering alone, it will be categorized as a Tampering threat and if data tampering is possible due to an XSS Vulnerability[11], it will be considered as an XSS threat. The STRIDE Model categorizes threats as follows:

Spoofing – an attack on authentication whereby there is an impersonation of something or someone else.
For example, since session identifiers are incremental, it is possible to guess what another user's session ID will be and generate this session ID in order to impersonate the user.

Tampering – an attack on integrity whereby there is modification of data.
For example, database entries can be modified using SQL injections.

Repudiation – an attack on non-repudiation whereby one claims to not have performed an action. For example, a system that does not have an audit functionality to monitor user operations in order to trace improper requests.

Information Disclosure - an attack on confidentiality by exposing information to someone not authorized to see.
For example, error messages that reveals the database schema.

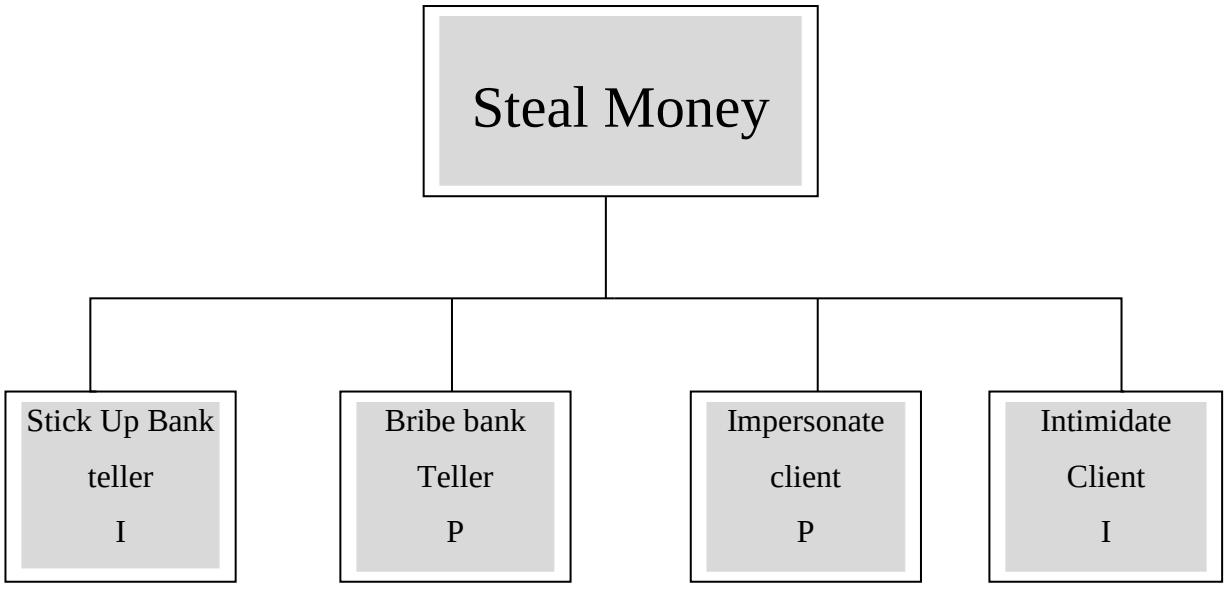
Denial of service – an attack on availability where there is a denial of service to users.
For example, system crashes from unexpected input.

[11] XSS is a cross-site scripting attack

Elevation of Privilege: an attack on the authorization policy whereby an attacker gains capabilities without proper authorization. For example, a user is able to get administrator rights through variable changes using buffer overflow attacks.

5.7 ATTACK TREES

Attack trees are methodical ways of describing the security of a system. According to Bruce Schneier [33] the attacks are represented with a tree like structure that starts with a root/parent node. This root node has one or more leaf/child conditions that must be true for an attacker to exploit a threat. In turn, any of these child conditions may have one or more children of their own. For example, an attacker may want to steal money from a bank, he/she then plans out an attack tree like the one shown in Figure 24, of how he/she is going to go about stealing the money, and label the different plans with either P where the attack is probable or I where the attack is improbable. Steal Money denotes the root/parent node and the rest denote the child nodes.



P – Probable

I – Improbable

Figure 24 : Attack Tree to steal money from a bank

Attack trees are quite easy to be constructed and flexible and give a great overview on the attacks that might be made to a system. These trees can be used to understand the process of security for an attacker. While the STRIDE model helps to identify and categorize threats and vulnerabilities, attack trees determine which threats should be addressed and in what order. This can be useful in understanding attack patterns that require events to occur in a certain sequence.

In addition to their usefulness in threat modeling, attack trees are widely used by penetration testers to create simulated attacks against systems by following the logical paths documented in the attack trees. Individual attack trees can also act as conditions for other trees, meaning that the condition of one tree involves the exploit of another tree. This is called *attack chaining* [12].

The next step in the threat modeling process is prioritizing vulnerabilities to determine the order in which attacks must be addressed and mitigated. The DREAD model, which will be discussed in the next section, will be used to prioritize these vulnerabilities of which the end result will be a sorted list from Highest impact (more severe) to lowest impact (less severe) vulnerabilities.

5.8 DREAD RATING

The DREAD model is a method of numerically evaluating the vulnerabilities of a system by assigning them scores or ranks [12]. A typical DREAD model would use a scale of 1 to 10 for the scores with 1 denoting the lowest (least dangerous) and 10 the highest which is the vulnerability that causes more danger. Once a vulnerability has been given a rating for each DREAD category, the ranks are summed up and averaged to find the vulnerabilities overall rating. The list is then sorted from highest to lowest prioritizing vulnerabilities that must be addressed.

Damage Potential: This is the amount of damage that could happen if the vulnerability should be exploited. This is not only a physical damage but it could also be an abstract damage such as reputation and loss of customer confidence.

Reproducibility: This is a measure of how easy or difficult it is to reproduce the attack. If vulnerability can be exploited on every attempt, then it would rate a higher score than an exploit attempt that is only successful at every 1000 attempts.

Exploitability: This is a measure of how easy or difficult it is to launch an attack. This includes factors like preconditions needed to carry out the exploit, how many steps are required, what is the difficulty in find a system that meets all the preconditions.

Affected Users: This is the measure of the total percentage of users that would be affected by this exploit. The greater the percentage of users affected the higher the rating.

Discoverability: This is the measure of how likely it is that a vulnerability will be found if it remains unpatched.

5.9 SUMMARY

In this chapter, we discussed the theory of threat modeling and the steps that could be taken to building a threat model. We looked at identifying assets that might be of interest to an attacker, decomposing the application through the use of DFD's, generating threats and classifying them using the STRIDE model. Attack tress were discussed which give a good overview of the process of security for an attacker. Lastly threats were ranked using the DREAD model.

CHAPTER 6

MICROSOFT THREAT AND ANALYSIS MODELING TOOL

6.1 INTRODUCTION

In this chapter, the theory on threat modeling learnt from the previous chapter will be applied to building a threat model for the Hackmi2 Social Network application using a threat modeling tool. This threat modeling tool will aid in providing the possible threats on the social network and the mitigation strategies. Before building the threat model, the "Microsoft Threat Analysis and Modeling Tool" will be discussed briefly.

6.2 THE TOOL

The Microsoft Threat and Analysis Modeling tool allows users to create and edit threat models for applications. This is to identify threats while facilitating the process of defining a security strategy. The tool is a Graphical User Interface based (GUI) .Net application and requires the Microsoft .Net framework version 2.0 to be installed. The threat model organises data points such as business objectives, user roles, components, external dependencies and application use cases into a tree-based view like one below, see Figure 25.

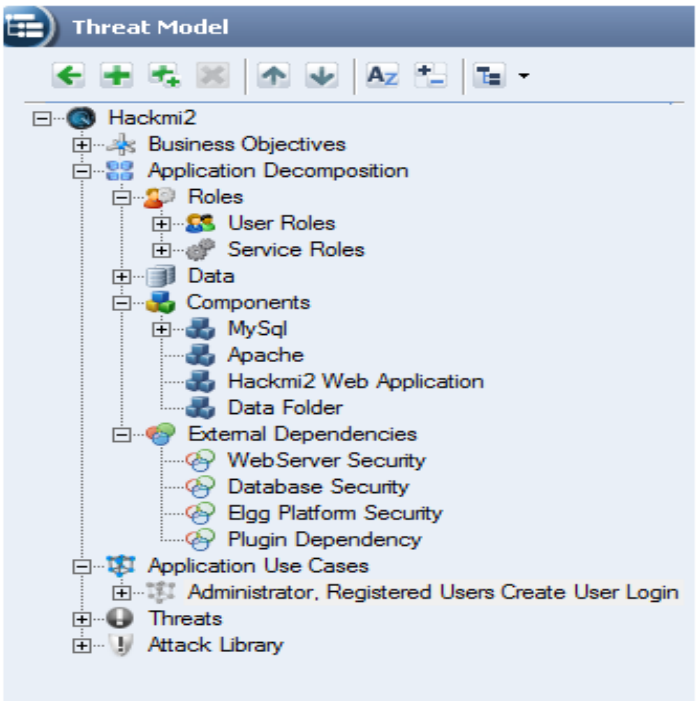


Figure 25: Tree-based view of threat model

Creating a threat model using Microsoft Threat and Analysis model tool is a three step process:

- **Defining the application requirements** is the first step in creating a threat model; this includes business objectives why are we creating this threat model (Chapter 6.3.2); user roles and service roles (Chapter 6.3.3)

- **Characterising the security of the system** looks at the building blocks and internal workings of the application that are security critical. This step looks at identifying implementation specific threats through the defining of components, External dependencies (Chapter 6.3.3) and Application use cases (Chapter 6.3.5). The threats identified will be categorized using the STRIDE model which the threat modeling tool uses to classify and categorize threats.

- **Risk measurement and mitigation of threats** is the last step when using this tool whereby after identifying the threats (Chapter 6.3.7) and categorizing them using the STRIDE model, the threats are analysed to check if they are mitigated. Once these phases are complete, the threat model can be assimilated using visualizations and reports. The tool automatically generates threats associated with an application based on the information provided.

The Threat and Analysis model can produce valuable security artifacts such as:

- Data access control matrix
- Component access control matrix
- Subject-object matrix
- Data Flow diagrams
- Call Flow diagrams
- Trust Flow diagrams
- Attack Surface
- Focused reports
- Customizable reports

6.3 THREAT MODELING IMPLEMENTATION

6.3.1 Threat Model Information

Table 3 shows the basic information about the Name and Description of the Social Network. In this modeling process, the threat model will be conducted on the Hackmi2 Social Network.

Threat Model Information	
Name	Hackmi2 Social Network
Description	Hackmi2 is a Social Network that combines elements of blogging and file sharing. The Social network has features such as chatting, updating of status, inviting friends and creating public or private groups.

Table 3 : Threat modeling information

6.3.2 Business Objectives

Business objectives are the goals for wanting to attack the Social Network; they describe the goals behind the attacks of the social network see Table 4. These objectives are supported by use cases.

Business Objectives
Exploit SQL Vulnerabilities on the Hackmi2 Social Network in order to delete the database
Exploit Cross-Site Scripting Attacks on the Hackmi2 Social Network in order to deface it
Look for other loop holes in the security of the system and exploit these
Create Antidotes for found vulnerabilities in order to close them

Table 4 : Business objectives

6.3.3 Application Decomposition

Roles

Roles define the levels of privilege users have. They are logical groups of users who use the Social Network. These groups can be divided into two sub-groups, namely, User Roles and Services Roles. User Roles involve users who interact with the social network whilst Service Roles contain users under which services are contained. Service roles are also known as trust levels or trust boundaries. All roles have to be authenticated by a mechanism depending on what they are doing (Administrator logging into the social network via a web interface will be authenticated using forms whilst if they login to the server directly, they will be authenticated using SSH). See Table 6.

User Roles			
Name	Description	Auth. Mechanism	# of Identities
Administrator	This is the owner of the social network and has access to all the functionality of the social network. The administrator has root access to the social network web server and database.	Forms	1-5
Registered Users	These are users who are registered and have a profile on the social network.	Forms	101-500
Unregistered Users	These are potential users who would like to use the social network but have not yet registered.		

Table 5: User Roles

Service Roles			
Name	Description	Auth. Mechanism	# of Identities
Website Role	To Authenticate users to the social network	Forms	1-5
Webserver Role	To authenticate Administrators via SSH	SSH Authentication	1-5
Database	To Authenticate Web Application accessing the database	Database Authentication	1-5
Physical Storage Role	To authenticate Administrators who will have physical access to the Server machine running the social network.	Access to Server Room	1-5
Operating System Role	To Authenticate Administrators to the Server Operating System.	SSH Authentication	1-5

Table 6: Service Roles

Components

Components are the high level building blocks of the social network. Components are logically broken down into services and objects with which the users interact with the system. Both services and objects can communicate in order to fulfil a user action. Component need to specify the roles that are going to interact directly with the component, the type of component, the technology its using and the action of the component on how it should behave. See Table 7.

Components				
Name	Roles	Type	Tech. Type	Run As
MySQL Database	Administrator Database	Database	MySQL Server	Database Process Identity
Objects				
Name	Description			Roles
Elgg_Table	Stores users data including user names, password and email. Plugin information is also stored such as plugin ID's.			Database
Apache Webserver	Administrator Webserver Role	Web Service	Apache 2	Webserver Process Identity
Hackmi2 Web Application	Administrator Registered Users Unregistered Users	Website	Elgg Social Network Engine	Website identity
Data Folder	Webserver Role	File Service	Ubuntu Server	Physical Storage Identity
Ubuntu Server	Administrator Webserver Role	Operating System	Ubuntu Server	Operating System Identity

Table 7 : Components

External Dependencies

External dependencies are components which are external to the Social Network and the social network does not have control over the behaviour or the implementation of these components. They also have to specify the type of dependency they have as seen in Table 7.

External Dependencies		
Name	Description	Dependency Type
WebServer Security	Hackmi2 Social Network depends on the security of the Apache webserver	Web Service
Database Security	Hackmi2 Social Network depends on the security of the MySQL Database Management System.	Database
Elgg Platform Security	Hackmi2 depends on the security of the Elgg platform. If the Platform is compromised, successful attacks might compromise the system as a whole.	Other
Plugin Dependency	Some plugins like PHPMailer rely on other applications like mailserver/	Other
Users	In order for the social network to be effective there needs to be users.	Users
Administrator	In order for the Social network to be functional at all times, it requires administrators.	Users

Table 8: External dependencies

Data

The data can be decomposed into different data element groups based on classification. Data is stored in one or more components (e.g. user names and passwords are stored in a database) which are then accessed by a user. Since the Hackmi2 social network is a huge system with a lot of data element groups, for conciseness we will only show some of the data elements that are in the social network.

Table 9 shows some of the data elements that were inserted into the Microsoft TAM. The Microsoft TAM allows data to be rated using six scales which are:

- High Business Impact Personally Identifiable Information (PII)
- Medium Business Impact PII
- Low Business Impact PII
- High Business Impact
- Medium Business Impact
- Low Business Impact

Personally Identifiable Information (PII) refers to data that can be used to uniquely identify, contact a user or can be used with other sources to uniquely identify a single individual. For example, a users profile contains their display name, profile picture and email address and that is a high impact PII because an attacker has all the information that uniquely identifies a user. The attacker might use that kind of information to carry out other malicious attacks on a single user my sending phishing scams via email. On the other hand plugins data elements have a High Business Impact since the functionality of the social network depends on them, but are not PII simply because they do not pose any danger of exposing user information on their own. TAM allows four access control levels on data which are: create, read, update and delete. Roles can be assigned on who has access to that type of data and what permissions they have. Conditions on certain permissions can be specified e.g. normal registered users can update a profile but only their own personal profile.

Data

Name	Description	Data Elements	Data Classification	Data Stores
User Profile	This is the e-Portfolio of the user which describes who they are	User Display Name Profile Picture User email address	High Business Impact PII	MySQL Database Elgg_Table

Access Control

Role	Access Control				Condition
Administrator	C	R	U	D	can create profile for users who have problems with email validation
Registered Users	C	R	U		in case they want to reopen their profile

Form Login	Username as Password Credentials	Username Password	High Business Impact PII	MySQL Database Elgg_Table

Access Control

Role	Access Control				Condition
Administrator	C	R	U	D	
Registered Users	C	R	U		Only on personal login details

User Registration		User display name Email address Username Password	High Business Impact PII	MySQL Database Elgg_Table

Access Control

Role	Access Control				Condition
Administrator	C	R	U	D	
Unregistered Users	C	R	U		

Plugins	Plugins for The Social Network	Blogs The wire Bookmarks Site-wide categories file browser database garbage collector Groups Log Browser Members Message Board Notifications TinyMCE 3 Column River Activity HTML Purifier	High Business Impact	Ubuntu Server

Table 9 : Data elements

6.3.4 Data Flow Diagrams

The concept of Data flow diagrams was discussed briefly in chapter 5.5.3 as being useful in providing a detailed representation of the social networks components, complete with boundaries and connections. In this section we will use the above information (Roles, Components, and External Dependencies, data) to construct a DFD in order to have a visual representation of the Hackmi2 Social network. The Different components of DFD are explained below by Table 10.

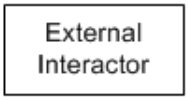
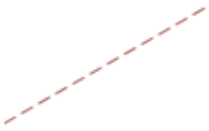
Component	Description
	Processes are data transforms. They transform input flows into output flows
	Same as above but instead of a single process, they are multi-processed meaning they consist of more than one process.
	External interactors are components that interact with a business process on the DFD but fall outside of the boundaries of the DFD
	Data stores represent information (i.e. data or control). Information can be read from a store and written to a store.
	Data flows define the interfaces between the components within the system and its external components.
	These are the trust boundaries which indicate the level of access of users, processes and various other components.

Table 10: DFD Components

A picture is worth a thousand words, and threat modeling is no exception. After decomposing the social network and looking at how the different components interact with each other, the following DFD (see Figure 26) was produced in order to have a visual representation of the system. The DFD shows three types of users, namely, unregistered users, registered users and administrators. The Red line show the trust boundary meaning if the users would like to login into the server, they first have to go through some authentication. Process boundaries are trust levels between processors and data stores (green) and machine boundaries are trust levels between two separate machines, for example, plugins communicating with external plugin dependency which can be located on another machine (see Figure 26).

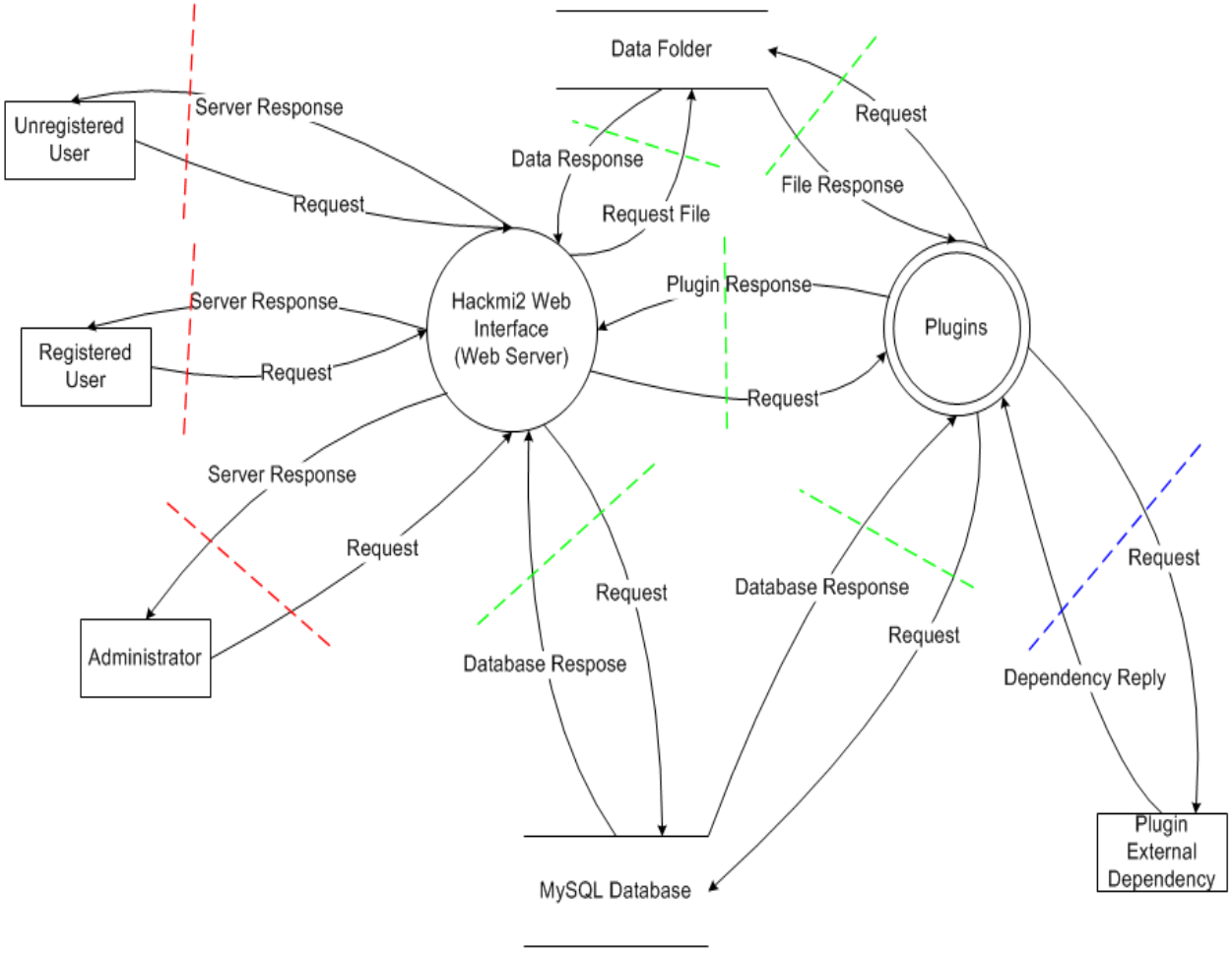


Figure 26: Data flow Diagram for the Hackmi2 Social Network

6.3.5 Application Use Cases

Use cases are fulfilled by a sequence of calls. Each call captures the user's action on a particular component. Brainstorming on how the system is intended or not intended to be used helps put boundaries around the threat modeling. The Microsoft TAM tool automatically creates use cases and they are based on the way the social network was modelled. Below are the some of the use cases generated automatically by the tool, see Table 11.

Application Use Cases			
Name	Description	Roles	Net Data Effect
Administrator, Registered Users Create User Profile		Administrator Registered Users	**C** User Profile
Administrator, Registered Users Read User Profile		Administrator Registered Users	**R** User Profile
Administrator, Registered Users Update User Profile		Administrator Registered Users	**U** User Profile
Administrator Delete User Profile		Administrator	**D** User Profile
Administrator, Registered Users Create Form Login		Administrator Registered Users	**C** Form Login
Administrator, Registered Users Read Form Login		Administrator Registered Users	**R** Form Login
Administrator, Registered Users Update Form Login		Administrator Registered Users	**U** Form Login
Administrator Delete Form Login		Administrator	**D** Form Login
Administrator, Unregistered Users Create User Registration		Administrator Unregistered Users	**C** User Registration
Administrator, Unregistered Users Read User Registration		Administrator Unregistered Users	**R** User Registration
Administrator, Unregistered Users Update User Registration		Administrator Unregistered Users	**U** User Registration
Administrator Delete User Registration		Administrator	**D** User Registration
Physical Storage Role, Administrator Create Plugins		Physical Storage Role Administrator	**C** Plugins
Physical Storage Role, Website Role, Administrator, Database Service Role Read Plugins		Physical Storage Role Website Role Administrator Database	**R** Plugins
Physical Storage Role, Website Role, Administrator Update Plugins		Physical Storage Role Website Role Administrator	**U** Plugins
Physical Storage Role, Administrator Delete Plugins		Physical Storage Role Administrator	**D** Plugins

Table 11: Use Cases for Social Network

Each Use Case has a call which is a coupling of a caller for a specific action. This is where we can specify data sent or data received by the caller during the call.

A collection of calls for a specified action allows you to define how that use case is realized in the context of your application. Recall that use cases define the set of actions or features that need to be supported by your application, and these allotted actions can be executed by specific roles in order to achieve a net data effect. Put simply, use cases define what needs to happen, and calls define how it happens.

Consider a Use Case where an administrator wants to do an SSH login into the social networks server machine in order to perform some maintance. A simple call action that can be performed would be **Administrator-> SSH login to webserver** where the administrator sends SSH login data to the social networks server in order to login. Microsoft TAM has three types of visualizations for calls, namely call flow, data flow and trust flow graph. Basically these graphs show how elements such as (roles, data and components interact). Figure 27 shows the call flow diagram for an administrator doing an SSH login to the server.

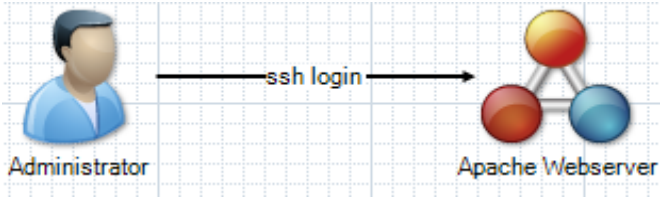


Figure 27: Call Flow Diagram for SSH Login

The data flow basically shows how data is transmitted when the administrator performs a SSH login.

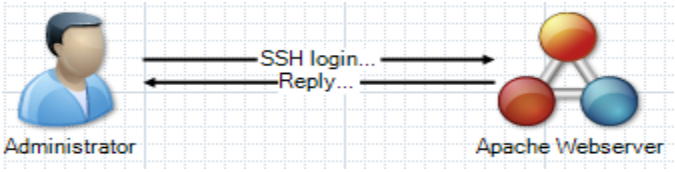


Figure 28: Data Flow Diagram for SSH Login

The trust flow shows the types of roles that are performing a call.

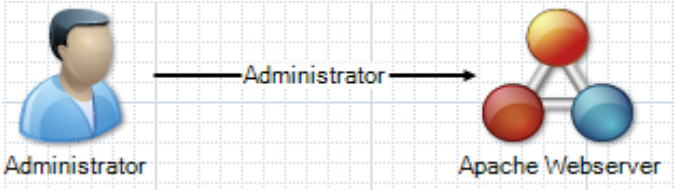


Figure 29: Trust Flow Diagram for SSH Login

Consider another case where administrator registered users and unregistered users would like to login into the Social Network. Figure 30, 31 and 32 show the call flow, data flow and trust flow respectively of such a Use Case. In this case all users will have to make a call to the Hackmi2 Social network by logging in using a form in which they enter their usernames and passwords. The web application will then send the requests to the server which in turn looks up the database to check if the user is authorized to login (see Figure 30).

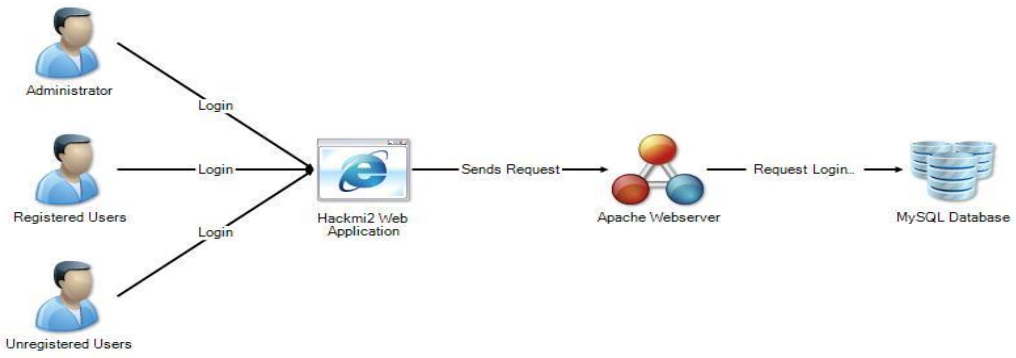


Figure 30: Call Flow Diagram

The data flow call Diagram (Figure 31) shows the data flow between the various components, what is interesting is that there is a dotted line for the unregistered user which denotes that no data was sent nor received. This comes as no surprise because in essence, unregistered users cannot login into the social network unless they have registered first.

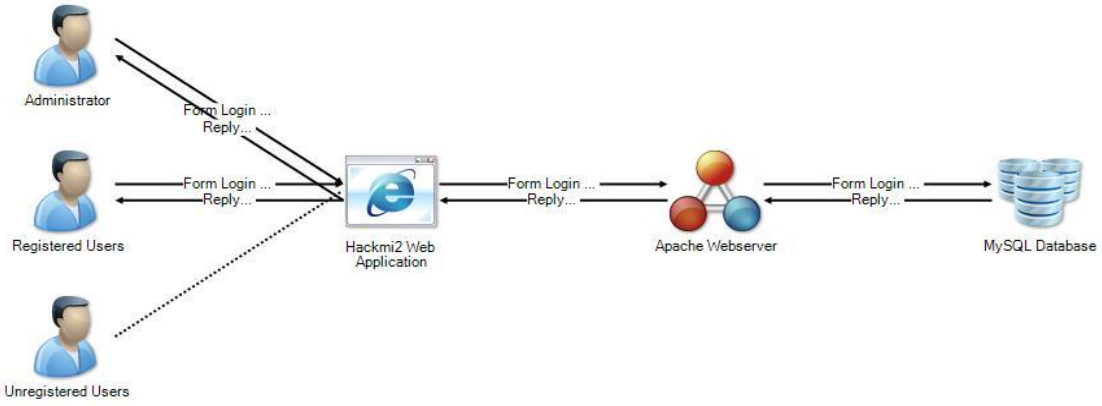


Figure 31: Data Flow Diagram

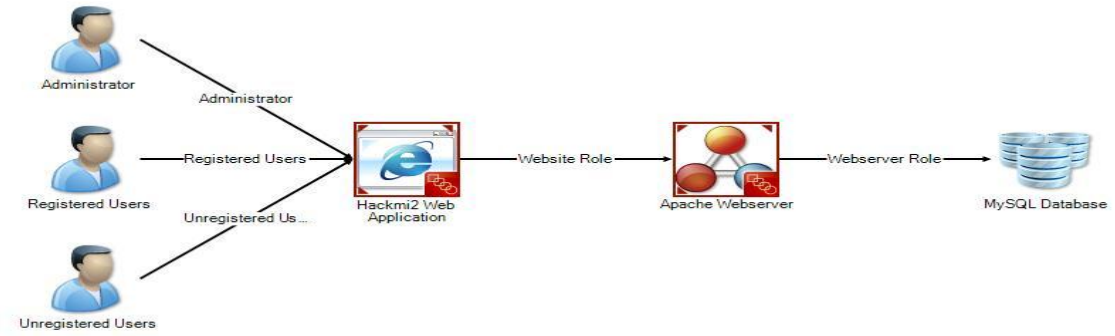


Figure 32: Trust Flow Diagram

6.3.6 Threats generated

Using the above use cases and calls, the Microsoft TAM automatically generated threats that might serve as vulnerabilities on the social network, see Figure 33. The Modeling classifies threats using the STRIDE model but groups them under three categories, namely Confidentiality where only authorised parties should have access to sensitive data , Integrity where the data should remain accurate and uncorrupted, and Availability where data must be available at all times as depicted in Figure 33. Note these are the threats associated with the two use cases that were discussed above.

Figure 33: Threats Generated by the tool

6.3.7 Attack Trees

The Microsoft TAM has the ability to generate attack trees but these tend to be very large. Figure 34 shows only a part of the attack tree. The attack three shows the Vulnerabilities, their cause and mitigation strategy. For example, Cross-Site Scripting can be caused by Ineffective encoding and validation as shown on Figure 34. The mitigation strategies for these attacks are performing context sensitive encoding and untrusted input should be validated against an inclusion list.

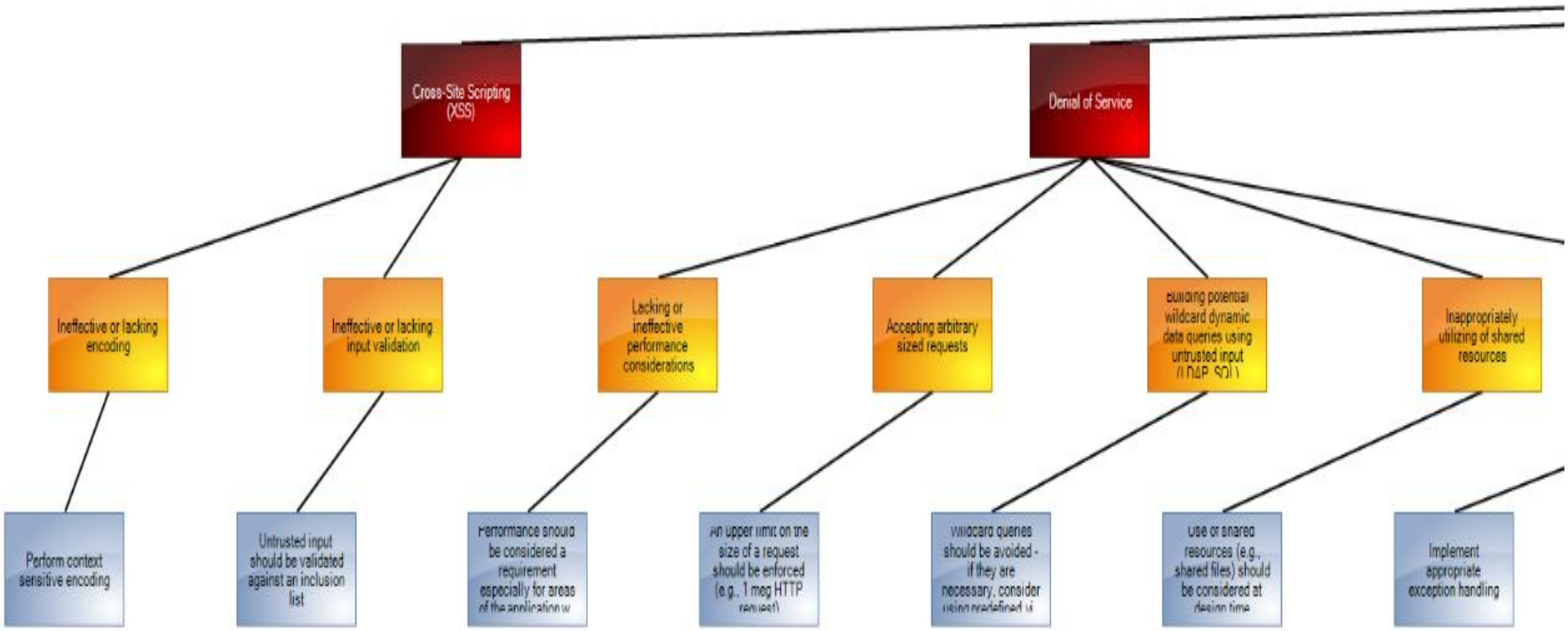


Figure 34: Attack Tree for hackmi2 Social Network

CHAPTER 7

COMPARING THREAT MODELS

In this chapter, we compare the three threat modeling tools in terms of their advantages and disadvantages in the threat modeling process.

7.1 MICROSOFT SDL MODEL

Advantages

The SDL threat modelling tool lets one create a high level overview of the system of application architecture using Data Flow Diagrams (DFD's). By just defining the data flow between the components of the application and pointing out the trust boundaries in the application landscape, this tool will help one point out the points where security attention is needed.

Microsoft SDL threat modelling tool is designed in such a way that even people who are not security experts can use it. Microsoft SDL threat modelling tool was designed specifically for use by any software architect since this tool embeds prescriptive, at-a-glance help throughout its use.

The Microsoft SDL threat modelling tool is centred on the software, as opposed to other threat modelling tools that centre on assets or attackers. It builds on activities that all software developers and architects are familiar with; such as drawing pictures of their software architecture.

Disadvantages

Microsoft SDL threat modelling tool uses Microsoft Visio to draw DFD's. A machine without the Microsoft Visio won't be able to run this tool, which means that before one starts using SDL threat modelling tool, although it is free, one has to pay for the license to use Microsoft Visio.

Since the SDL threat model uses the high level overview, sometimes there is a need to go in to more detail. For example, defining the technology used in building a component and specific threats for the technology. The quality of the resulting report depends on the knowledge of the one who created the model. This tool lacks the possibility of prioritising the threats. Not every threat is likely to happen (because of other factors) and not every threat has the same impact on different business. Threats with a higher priority demand more attention, while low priority threats can be left unattended, thus saving money or leaving user-friendliness intact.

7.2 SENSEPOST CTM

Advantages

The SensePost CTM Ranks threats according to their impact on the system using a risk equation

Risk = Applied Likelihood x Value at Risk
Applied likelihood = Attack likelihood which is reduced by the user trust and location trust
Value at Risk = Value of asset which is reduced by the amount of asset exposed by an attack.

The risk equation makes it easy for developers on deciding which threats to mitigate. The developer needs to define the threats they want to incorporate into the threat model, therefore the threat modeling tool will only recognize threats defined by the developer. A mapping of interfaces and threats makes it easy to see which threat will be available on which interface. The tool maps users and interfaces to locations making it intuitive to see which users will be available at a certain location and interface e.g. Users located on a public network through a web interface.

Disadvantages

The SensePost CTM is very subjective when it comes to modeling the application. The trust level of users, location and interface ratings are determined by the developer. If an interface is given a low rating it would affect threats that are available at that interface i.e. if anonymous users, a public network and web interface are given high trust then the threat report would rate the impact of threats on a public network on a web interface as insignificant. The SensePost CTM is designed to be used by a security expert, as you need to have knowledge about threats. The developer can oversee some threats as the CTM does not generate threats automatically i.e. if the developer does not have knowledge about the XSS attack, then they would not model it as one of the threats to the web application.

7.3 MICROSOFT TAM

Advantages

The core function of the Threat Analysis & Modeling tool is to identify threats, while facilitating the process of defining a security strategy. Even if you are not a security subject-matter expert, you have the ability to consistently and objectively identify threats to your software application.

Users can assimilate threat models through analytics, visualizations, and reports.

The Threat Analysis & Modeling tool automatically generates potential threats to your software application, based solely on known information that you provide. The Threat Analysis & Modeling tool also has the capability to assimilate the information you provide to build security artifacts such as access control matrices, data flow and trust flow diagrams, and focused, customizable reports.

An attack library is a collection of attack types along with their relevant vulnerabilities and proposed countermeasures to those vulnerabilities. Attack libraries enable software application teams to define and adopt secure engineering techniques, gain the information necessary to detect security concerns, and create relevant security test cases. Attack libraries provide a way to define, with absolutely minimal permission, the relationship between the exploit (attack), the cause (vulnerability), and the fix (countermeasure). The attack library helps ensure that various development teams understand the security assumptions and dependencies of your application

Disadvantages

Trees generated are huge and cannot be visualized properly using a normal computer monitor. The Modeling tool is not as intuitive to use like the Microsoft SDL tool which uses DFD’s to model components and their interactions.

CHAPTER 8

VULNERABILITY CASE STUDY

8.1 CROSS-SITE SCRIPTING

Cross-Site scripting (XSS) is a vulnerability rooted in the way HTML content is generated and interpreted by a browser. These Vulnerabilities are usually found in web applications and allow attackers to inject HTML code into the input of a web-based application. This basic vulnerability, depending on how it is carried out, can be used for a multitude of purposes which include: Cookie theft, Session ID theft, defacing web site, cookie poisoning and Denial of Service attacks.

We carried out an XSS attack on the Hackmi2 Social network to see if it was vulnerable to XSS injections.

Procedure:

We used TamperIE which is a browser extension for HTTP analysis and tampering. This tool allows you to tamper with GET and POST request. By default it is set to tamper only with POST's, so when you encounter a POST while browsing (such as a form submission), TamperIE intercepts the submission and presents the screen shown on Figure 35.

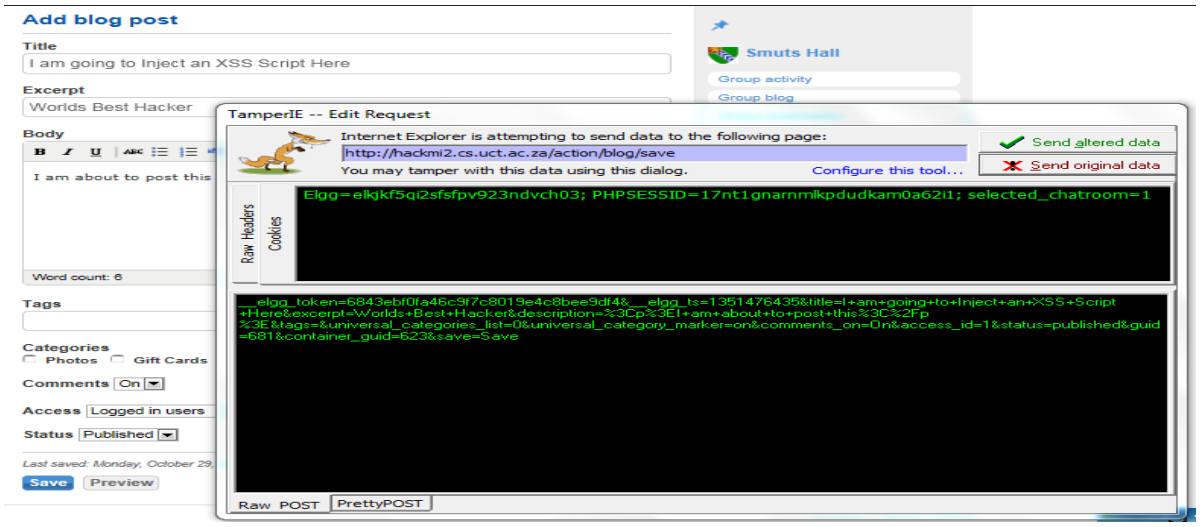


Figure 35: TamperIE

With respect to Figure 36, we selected the "PrettyPOST" tab which shows us all fields we can tamper with. In this case we are particularly interested in Tampering with the description field. We insert simple Java Script that just pops out an alert box written "You Got Hacked!!! by UCT Compsci".

Script: *'"<script>alert ("You Got Hacked!!! By UCT Compsci") ;< /script>*

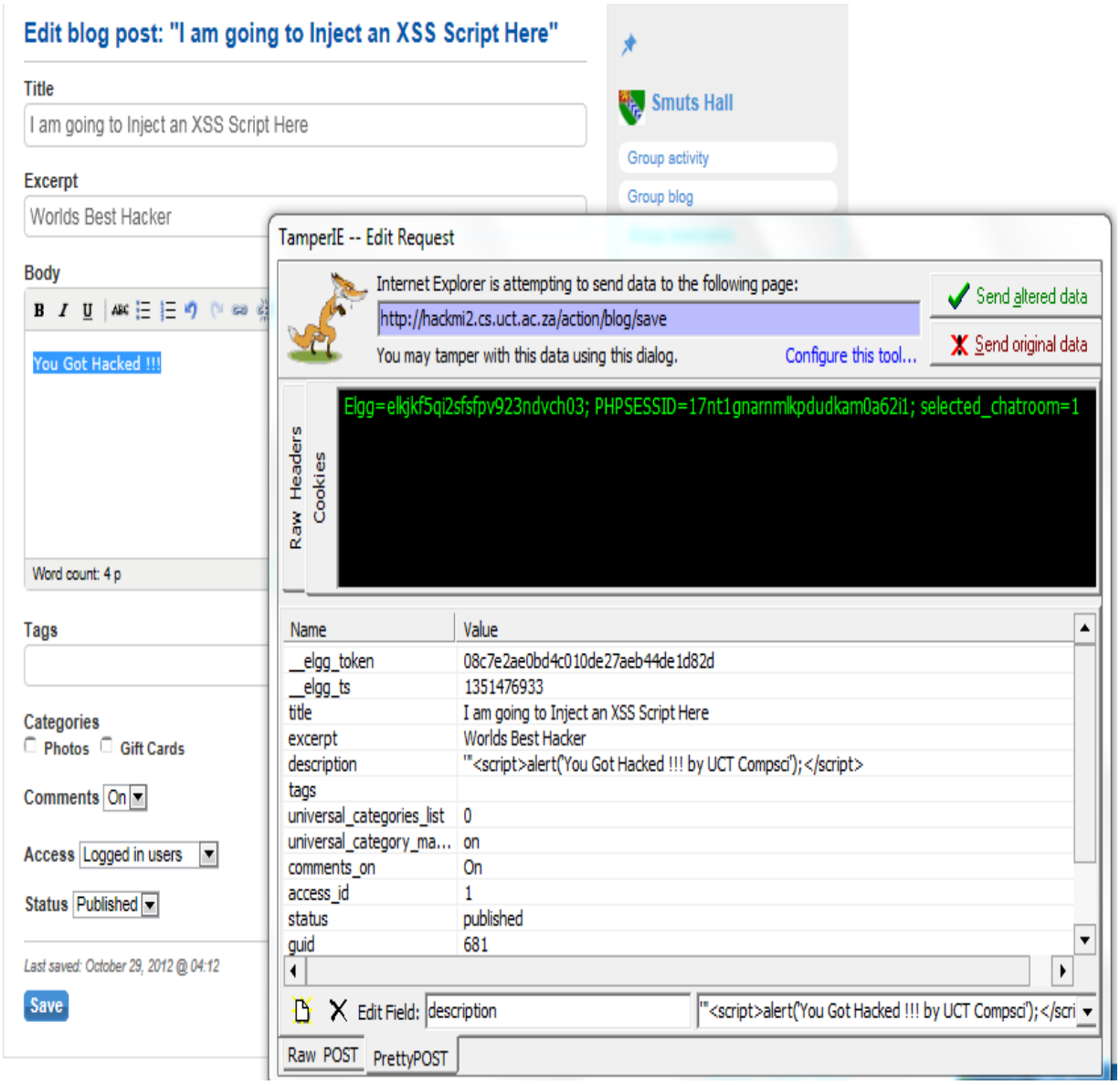


Figure 36: Injecting XSS Script

Figure 37 shows the Result of the injection to show that the attack was successful.

Figure 37: XSS Vulnerability

8.2 DENIAL OF SERVICE

The above example might not seem dangerous but consider if the script was modified by just adding an extra Java script line: *Location.reload()* therefore modifying the injected script to

'"<script>alert ("You Got Hacked!!! By UCT Compsci") ;< /script>
Location.reload()

This will keep on generating alert boxes continuously as a victim clicks on the OK button. This process will not stop unless the victim completely reboots his machine. This type of attack is known as a denial of service attack where the victim as a legitimate user of the social network is denied access from social networking services even if it is temporary.

8.3 COOKIE THEFT LINK ATTACK

The following code will steal a cookie from a user if they click on a malicious link. The link will steal the victim's cookie and stores it in the attacker's machine: cheetah.cs.uct.ac.za

<a
href="javascript:document.location='http://cheetah.cs.uct.ac.za/logger.php?cookie='+document.cookie;">CLICK ME!

The following script also does the same thing but it is crafted more elegantly as it does not rely on the user clicking a link but on the mouse action event, that is, if the user moves the mouse, the attack will be carried out.

Below is the code that executes this attack:

style="position:absolute;top:0px;left:0px;height:100%;width:100%;display:block;z-index:400" onmousemove="document.location='http://cheetah.cs.uct.ac.za/logger.php?cookie='+document.cookie;

8.4 COUNTERMEASURE

This a java script patch that can be used to sanitize input so to block XSS scripts that might lead to Denial-of-Service attacks. This code also blocks cookie thefts by sanitizing the users input.

Javascript block

```
<script language="javascript">
        var id = <%=AntiXss.JavaScriptEncode(Request.Querystring["userinput"]) %>;
</script>
```

8.5 SQL INJECTIONS

SQL injections were unsuccessful on the social network because the Elgg framework has done a very good job of sanitizing SQL queries.

CHAPTER 9

CONCLUSIONS

The application of threat modeling and comparison of threat modeling tools to social networks proved to be feasible. In order to demonstrate this feasibility, our first task was to implement a Social networking site that allowed users to register, login, create profile chat and blog. In order to do this, we decided to use the Elgg open source social networking engine and over a period of two months, we developed the social network.

This proved to be a challenge as we did not have any prior experience in web programming. We had to install and configure the server machine ourselves as part of the learning process. After finishing and deploying the social network, we invited friends and family to join. Today Hackmi2 is a fully functioning social network running on the UCT network with the domain http://hackmi2.cs.uct.ac.za and boasts a membership of approximately 200 users and it is still growing.

The next step in our project was to use threat modeling tools to model the security of the social network. This was not as trivial as we first thought; the main challenges were in reading and understanding the relevant literature, and changing mind sets and thinking like attackers. We needed to understand what attackers target when hacking social networks. For this we used the threat modelling process whereby we identified assets that attackers might be interested in, decomposed the application into different components using DFD's. The Microsoft TAM then classified threats using the Microsoft STRIDE model and rated them using the DREAD model.

The Microsoft TAM, Microsoft SDL and SensePost CTM were then compared qualitatively each with its own disadvantages and advantages. The SensePost CTM was more suitable for security experts as it did not generate any threats but you would have to define your own threats. The focus of the SensePost CTM was on providing decision making information. The Microsoft SDL model was more suitable for software–centric modeling approach than attack centric as it was designed for threat modeling before and during the application development.

The Microsoft Threat and Analysis Modeling (TAM) tool proved to be very useful in generating automatic use-cases and threats according to components specified. The tool is designed so that even non-security experts are able to model application security. Unlike the Microsoft SDL and SensePost CTM, The Microsoft TAM generates attack trees and mitigation strategies for certain vulnerabilities found.

The last part of the project, we tested out if it was possible to exploit the threats the Microsoft TAM exposed. This was successful as pointed out by the Threat Model. The Hackmi2 application was vulnerable to Cross-Site Scripting (XSS), Cookie Theft, and Denial-of-Service attacks. SQL injections proved to be very difficult to exploit because the Elgg framework has tightly secured against SQL injection by sanitizing input. We ended of by providing mitigation strategies for the vulnerabilities.

The project was very successful as we met all our aims which were to create a social network, Use a threat model to model and analyse threats in the social network, test the threats by hacking the social network, then providing countermeasures to the attacks.

9.1 FUTURE WORK

Future work in this project could be looking into exploiting more vulnerabilities in the Hackmi2 social network and protecting against them and then generalising them to other systems.

Another thing we could look at is moving the social network into a mobile platform so users can enjoy using the social network on their mobile phones.

We would like to see Hackmi2 getting more users and expanding as a social network of choice for UCT students.

REFERENCES

[1]. **Linden, M.A.Van der.** *Testin Security Code.* s.l. : Auerbach Publications, 2007.

[2]. **D.Tosh, B.Werdmuller.** Creation of a learning landscape: weblogging and social networking in the context of e-portfolios. [Online] 1999.
URL:http://www.eradc.org/papers/Learning_landscape.pdf.

[3]. **A.Campbell, R.Ammann, B.Dieu.** ELGG-A personal learning landscape. [Online] 2005.
URL: http://www.tesl-ej.org/wordpress/issues/volume9/ej34/ej34m1/.

[4]. **S.Macanda.** *Hackmi2.* [Online]2012. URL: http://www.hackmi2.cs.uct.ac.za.

[5]. Plugins/Blogs. *Elgg.* [Online] 2012. URL: http://blog.elgg.org/

[6]. BlogPulse. [Online] The Nielsen Company, 2011. URL: http://www.blogpulse.com.

[7]. **S.Bowen.** Blog writing : Easy as butter and hard as diamond. *webgranth.* [Online] 2012.
URL: http://www.webgranth.com/blog-writing-easy-as-butter-and-hard-as-diamond.

[8]. Market Share. *MySQL.com.* [Online] 2012.
URL:http://www.mysql.com/why-mysql/marketshare/.

[9]. RDBMS. *webopedia.* [Online] 2012.
URL:http://www.webopedia.com/TERM/R/RDBMS.html.

[10]. PHP. *PHP.net.* [Online] 2012. URL: http://php.net/manual/en/intro-whatis.php.

[11]. Using a Three-Tier Architecture Model. *MSDN.Microsoft.com.* [Online] 2012.
URL: http://msdn.microsoft.com/en-us/library/windows/desktop/ms685068(v=vs.85).aspx.

[12]. **D.White.** Sensepost Threat Modeling Metricon 6, Usenix Workshop. [Online] 2011.
URL:http://www.slideshare.net/sensepost/corporate-threat-modelling.

[13]. **J.Scambray, M.Schema,C.Sima.** *Hacking Web Applications Exposed 2nd Edition.* 2006.

[14]. **B.Schneier.** *Attack Trees: Modelling Security Threats.* [Online] Dr Dobbs Journal, 1999. URL: http://www.schneier.com/paper-attacktrees-ddj-ft.html.

[15]. **J.D. Meier, A.Mackman, M.Dunner, S.Vasireddy, R.Escamilla and A.Murukan.** *Improving Web Application Security: Threats and Countermeasures.* [Online] 2003. URL: http://msdn.microsoft.com/en-us/library/ff648644.aspx.

[16]. **N.Sportsman.** Threat Modeling . [Online] 2011. URL:http://www.praetorian.com/presentations/Praetorian_Threat_.pdf

[17]. **S.Burbeck.** Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller(MVC). *Illinois Department of Computer Science.* [Online] 1992. st- URL:www.cs.illinois.edu/users/smarch/st-docs/mvc.html.

[18]. **Elgg Core Developers.** Engine/Views. *Elgg.* [Online] 2012. URL: docs.elgg.org/wiki/Views.

[19]. **Microsoft.** ASP.NET MVC Overview. *msdn.microsoft.* [Online]2012. URL:msdn.microsoft.com/en-us/library/dd381412(VS.98).aspx.

[20]. **J.Naakka.** How to Build Your Own Social Network. *Slideshare.net.* [Online] 2010. URL:www.slideshare.net/intunex/how-to-build-your-own-social-network-with-elgg-4987242.

[21]. **U.Erlingsson, B.Livshits, Y.Xie.** End-to-End Web Application Security. [Online] 2007. URL:http://static.usenix.org/event/hotos07/tech/full_papers/erlingsson/erlingsson_html/.

[22]. **Chien, E.** Malicious Yahooligans. [Online] 2006. URL:http://www. symantec.com/avcenter/reference/malicious. yahooligans.pdf.

[23]. **OWASP Foundation.** *The ten most critical web application Security risks* . 2010.

[24]. **MITRE Corporation.** Common vulnerabilities and exposures. [Online] 2007. URL:http:// cve.mitre.org/cve/.

[25]. **J.Walden, C.E.Frank.** Web Application Security Tutorial. *ACM.org.* [Online] 2007. URL:http://delivery.acm.org/10.1145/1290000/1289294/p77walden.pdf?ip=137.158.153.204&acc=PUBLIC&CFID=180476981&CFTOKEN=53778176&__acm__=1350877346_a23a9f32009b31ee4413524bdacd4c92.

[26]. **Z.Minchev, M.Petkova.** *Information processes and threats in social networks. A case study.* H.5, I.2.4, I.6.5, K.4.2, K.6.5, 1998, ACM Computing Classification System.

[27]. **Wikipedia.** Social circle. *Wikipedia the Free Encyclopedia* . [Online] 2012. URL:http://en.wikipedia.org/wiki/Social_circle.

[28]. **C.Laorden, B.Sanz, G.Alvarez, P.G.Bringas** *A threat model approach to threats and vulnerabilities in On-line social networks, in :A.Herrero et al.***.** s.l. : CISIS AISC, 2010. Vol. 85, pp. 135-12.

[29]. **I.Paul.** Use Of Weak Authentication Parameters Puts Information Security Of Over Million Individuals At Risk. *http://www.pcworld.com.* [Online] 2010. URL:http://www.articlesnatch.com/Article/Use-Of-Weak-Authentication-Parameters-Puts-Information-Security-Of-Over-Million-Individuals-At-Risk/1903183#.UITdPW83te8.

[30]. **D.M.Boyd, N.B.Ellison.** Social Network Sites: Definition, History, and Scholarship. *jcmc.indiana.edu.* [Online] 2007. URL:http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html.

[31].techradar.com. Facbook hack [Online] 2010.
URL: http://www.techradar.com/news/internet/facebook-hack-puts-public- .

[32]. **M.Chacksfield.** Facebook ‘hack’ puts public data into the public domain . *techradar.com.* [Online] 2010. URL : http://www.techradar.com/news/internet/facebook-hack-puts-public-data-into-the-public-domain-706396.

[33]. **Google.** recaptcha. *google.com.* [Online] 2012.
URL: http://www.google.com/recaptcha/captcha.

[34]. **G.Oppy, D.Dowe.** The Turing Test. *The Stanford Encyclopedia of Philosophy (Spring 2011 Edition).* [Online] 2011.
URL:http://plato.stanford.edu/entries/turing-test/.

[35]. **Casteele, S.V.** *Threat Modeling for web applications using STRIDE Model.* 2005.

[36]. **D.De Cock, K.Wouters, D.Schellekens, D.Singelee, B.Preneel**. *Threat modeling for security tokens in web applications.***.** s.l. : In proceedings of the IFIP TC6/TC11 International Conference on Communications and multimedia security, 2005.

[37]. **A.S.Sodiya, S.A.Onashoga, O.B Ajayi.** Toward Building Secure Software Products. *Jour;nal of issues in Information science and information technology.* [Online] 2006.
URL: http://informingscience.org/proceedings/InSITE2006/IISITSodi143.pdf.

[38]. **I.Griggs.** Browser Threat Modeling. [Online] 2004.
URL:http://iang.org/ssl/browser threat model.html.

[39]. **A.Shrivastava.** *An Approach to Web Application Threat Modeling.***.** 2008.

[40]. **S.A.Klein.** Position paper on voting system threat modeling. [Online] 2005.
URL :http://www.docstoc.com/docs/21974886/Voting-System-Threat-Modeling.

APPENDIX A.1

This privacy statement discloses the privacy practices for Hackmi2.

Information Collection and Use

Hackmi2 is the sole owner of the information collected on this site. We will not sell, share, or rent this information to others in ways different from what is disclosed in this statement. Hackmi2 collects information from our users at several different points on our Web site.

Web Site Registration

In order to use some features of this Web site, users must first complete the registration form. During registration, users are required to give their contact information (name and e-mail address). This information is used to contact users about the topics on our site for which they have expressed interest and to enable users to retrieve lost passwords.

Cookies

Hackmi2 uses cookies to remember if users have logged in while on our site. This allows web site users to avoid logging in more than once, thereby saving time. Users have the option of disabling or not accepting cookies by changing the preferences on their browsers. If users opt to disable cookies, they will still be able to use our Web site. However, they will not be able to use some functionality or post to the message boards. No personally identifiable information (e-mail address, name, etc.) is collected with the cookies that we set.

Web Statistics

We use IP addresses to analyse trends, administer the site, track user movement, and gather broad demographic information for aggregate use for reporting and sponsorship purposes. IP addresses are not linked to personally identifiable information.

Links

This Web site contains links to other sites. Please be aware that Hackmi2 does not claim any responsibility for the privacy practices of such other sites. We encourage our users to be aware when they leave our site and to read the privacy statements of each and every Web site that collects personally identifiable information. This privacy statement applies solely to information collected by this Web site.

Security

This Website is not fully protected and we encourage users not to post personal information or information that might lead to financial loss, this is a research based social network and and the creators will not take any responsibility for security breaches.

Updates

We may also send the user site and service announcement updates. Members are not able to unsubscribe from service announcements that contain important information about the service. We communicate with users to provide requested services and to discuss issues relating to their accounts via e-mail or phone.

Notification of Changes

If we decide to change our privacy policy, we will post those changes on our home page so our users are always aware of what information we collect, how we use it, and under what circumstances, if any, we disclose it. If at any point we decide to use personally identifiable information in a manner different from that stated at the time it was collected, we will notify users by e-mail. Users will have a choice as to whether or not we use their information in this different manner. We will use information in accordance with the privacy policy under which the information was collected.

APPENDIX A.2

Hackmi2: Terms and Conditions

Hackmi2 gives no warranty and makes no representation as to the content of this site or its accuracy and accepts no liability or any errors or omissions in it. Hackmi2 does not warrant that your use of the site will not infringe third party rights. Hackmi2 does not warrant that use of this site or materials downloaded from it will not cause computer virus infection or other damage to property. It is a condition of use of the site and the materials in it that use is at the user's own risk. Neither Hackmi2 nor any of the site's editors or contributors shall be liable for any loss or damages suffered as a result of any use of the site, including but not limited to direct loss, consequential loss and loss of profits.

This site includes links to other sites on the World Wide Web. We are not responsible for such sites and cannot vouch for the suitability or accuracy of their content. You link to them at your own risk.

Hackmi2 may monitor communications on the site but is under no obligation to do so. You must not post any material onto the site which is personal, defamatory, obscene or blasphemous, which infringes third party rights or which could in any other way give rise to criminal or civil liability in any jurisdiction. We shall have no liability for any such material. We may in our absolute discretion remove any material if in our view it falls or might fall within the foregoing categories or is otherwise inappropriate. We shall own any material you send to the site or to us shall not be obliged to treat any such communication as confidential and may exploit any such communication and its contents in such ways as we see fit.

We may change these conditions of use from time to time. You will be bound by changes even if you do not re-visit this page to re-read this notice.

APPENDIX B

This is the process of retrieving a lost password.

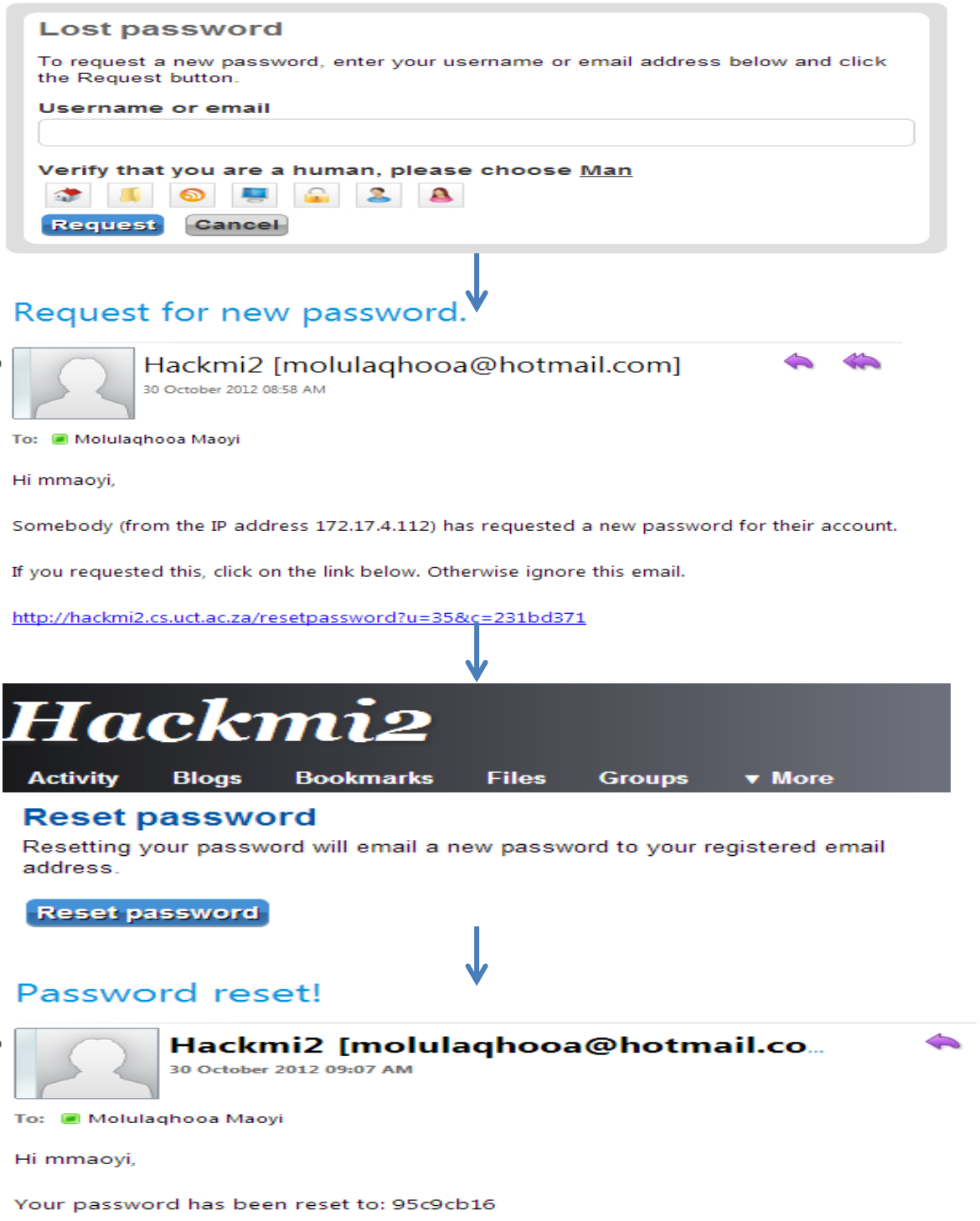


Figure 38: Retrieving a Lost Password

APPENDIX C

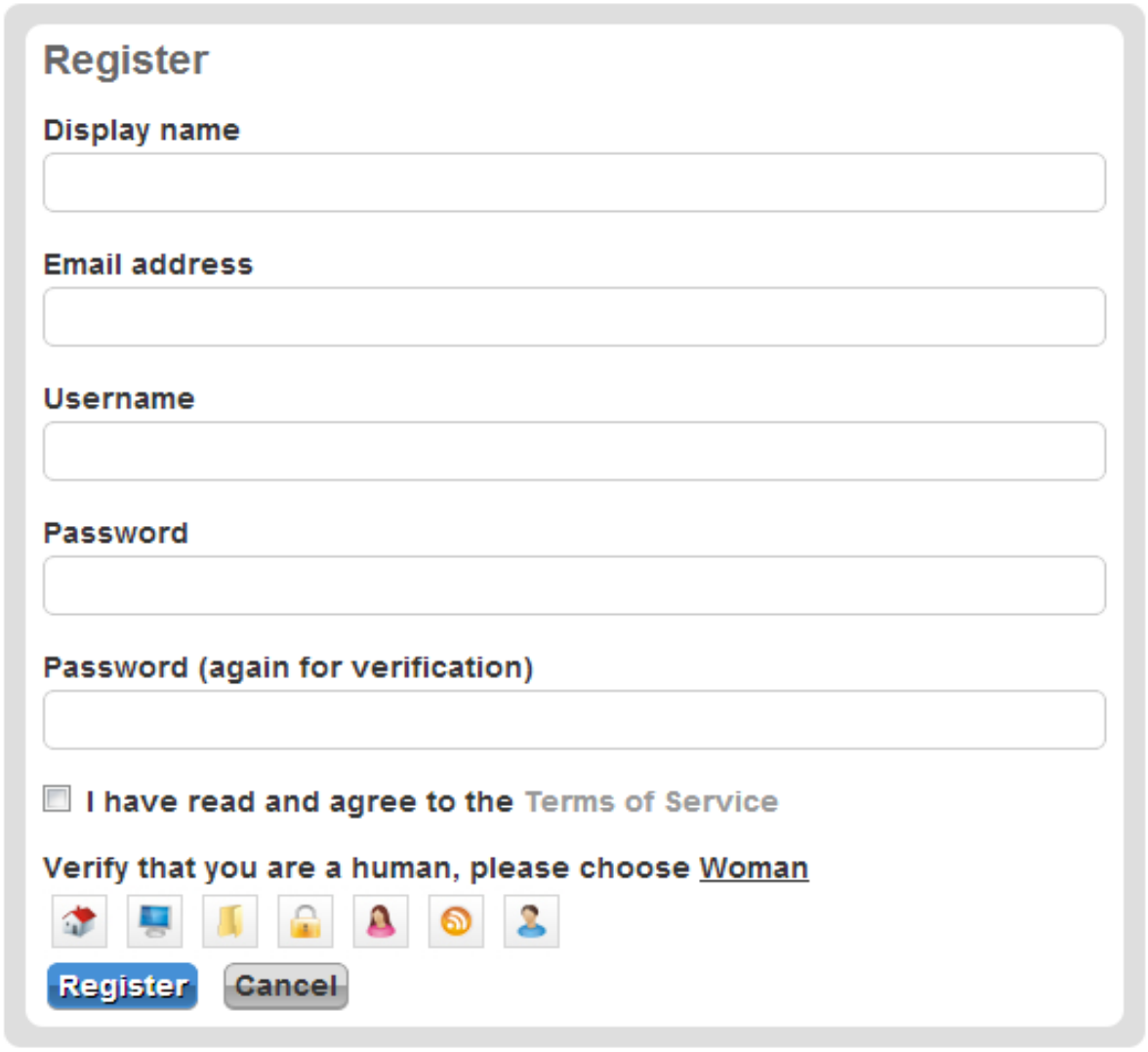


Figure 39: Registration Form