Automated Social Engineering Proof Of Concept

A case study of Facebook

MARKUS HUBER

Master's Thesis at DSV Seclab
This thesis corresponds to 20 weeks of full-time work.

2009-03

"While differing widely in the various little bits we know, in our infinite ignorance we are all equal." Karl Popper

I dedicate this thesis to my family, friends and my girlfriend Tasia, who unremittingly supported me during my years of study. David, Florian, Valentin, and Ingo did a fantastic job helping me with the proof reading of the thesis. I would also like to thank Andi for hosting the programming sessions and all the people who participated in the experiment. Furthermore I'm very gratefull for the help and support of the DSV SecLab staff. I would especially like to thank my supervisor Stewart Kowalski for his advice, Marcus Nohlberg for his expertise on social engineering, and Louise Yngström for showing me the beauty of academia. They all made this work possible.

Stockholm, March 2009
Markus Huber

Abstract

Automated social engineering (ASE) takes the classical social engineering attack one step further and makes it a time efficient and thus cheap attack. ASE is enabled through social networking sites (SNSs) which entail a pool of digitized personal information which make traditional social engineering approaches such as dumpster diving obsolete. We created a proof of concept ASE bot on the basis of Facebook which is one of the biggest SNSs at the time of writing. In order to evaluate the feasibility of ASE attacks on Facebook we conducted two experiments on the basis of our ASE bot implementation. In the first experiment we evaluated the information gathering functionalities of the ASE bot on basis of five Swedish multinational corporations. Although our application on average found more than eight possible targets per organization, the actual number was dependent on the organization's network size in Facebook and the privacy awareness of their employees. In the second experiment we performed a Turing test were twenty test subjects had to decide if they were talking to a real person or to the ASE bot. The test subjects in generally were able to identify the ASE bot with a high probability. Although Facebook has a number of protective measures in place the ASE bot did not get detected or blocked during our experiments simply because it aimed at simulating an average Facebook user. Our results in conclusion showed that ASE bots are feasible from a technical standpoint and that existing chatbots need to be adapted for social networking services.

Contents

List of Figures

List of Tables

Part I

Preface

Chapter 1

Introduction

1.1 Background

Social engineering is an attack on the security of systems, based on exploiting human factors. Victims are fooled into releasing information or performing a malicious action on behalf of the attacker. While the factor of success with social engineering is the deception of victims, attacks can involve technology as well (e.g. malware, Emails, manipulation of Software). Attacks go through different stages until a final goal is reached, and usually begin with collecting information on the future victims.

Social Network Sites (SNSs) like Facebook, LinkedIn, and MySpace account to todays most popular online services. SNSs users create online profiles and foster social relationships with each other. Social Networking Sites in any case are no longer just online playgrounds for teenagers; a growing number of employees are using them and are often encouraged by their management to do so. Users reveal valuable personal data like their full names, hometown, date of birth, and interests on their self created profiles. An attacker can use this information to perform sophisticated social engineering attacks. The social engineering attack can be automated because personal information is available in digital form and SNSs include functionality to communicate. Hence the emerging usage of these applications will change the methods perpetrators can use to exploit the behavior of possible victims towards automated tools.

1.2 Description of the research area

As social engineering is based on deception of humans, social sciences such as economics and psychology entail profound knowledge that helps to understand the used attack vectors better. Social networking sites on the other hand have been analyzed mainly regarding the privacy threats they state, but a variety of articles have been published recently on other security implications of these networks as well (e.g. Athanasopoulos et al. [2008], Hogben [2007]).
At the time of writing automated social engineering has barley been examined, with

Phishing being the closest research field. Especially worth noting is the contribution of Jagatic et al. [2007] on "Social Phishing", which makes use of data harvested from online social networks. Experiments on social engineering furthermore entail challenges regarding research ethics, Finn and Jakobsson [2007] discussed these issues on basis of Phishing experiments.

1.3 Research question(s)

We have done a study on the readiness for automated social engineering [Nohlberg et al., 2008], this thesis hence takes our initial results further in order to:

- Describe a probable automated social engineering attack.
- Develop a software application for automated social engineering based on Facebook's[1] online social network.
- Evaluate the feasibility of an automated social engineering attack on the basis of our proof of concept application.

1.4 Methodology

We used the *sophisticated falsificationist* approach as outlined by Kowalski [1994] as a research methodology:

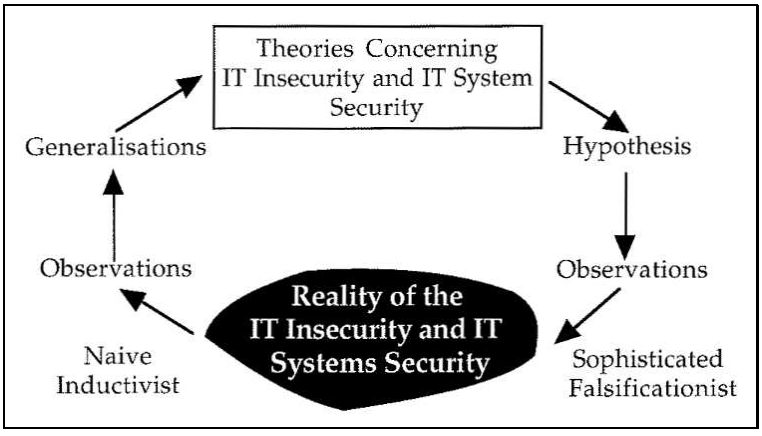

Figure 1.1. Naïve inductivist and sophisticated falsificationist [Kowalski, 1994]

Chapter 2 of this thesis is based on a literature survey in order to give an introduction and an overview to the fields of social engineering and online social networks. On basis of the body of acquired knowledge we hypothesized that automated social attacks can be implemented. Finally we made observations to test if

[1] http://facebook.com is one of the most popular Social Networking Sites at the moment.

our hypothesis reflects the reality of IT insecurity and IT systems security today. The observations are summarized in part II & III and consist of a proof of concept implementation (see chapter 4) and two experiments on basis of our software (see chapter 5).

1.5 The goal and purpose of the research

The goal of this research is to shed light on the rather unexplored area of automated social engineering within social networking sites to finally create: an introduction to this field, a sample ASE bot[2] for Facebook and an empirical study on the feasibility of such a bot. Thus the fulfilment of this goal would answer to which degree social networking sites render automated social engineering possible and how feasible Automated Social Engineering attacks are made. Except from its scientific contribution, this thesis intends to create awareness amongst security professionals in the public and private sector.

1.6 Limitations

As mentioned before social engineering includes findings from many different research areas which could be analyzed in depth, through an extensive literature study. The part on social engineering in this thesis is limited to a brief introduction and essentially the six social-psychological principles of influence from Cialdini. The ASE bot requires some form of artificial intelligence / learning system in order to communicate with possible victims which are rudimental due to the time constraints of this thesis.

1.7 Audience

The audience targeted for this thesis are primarily information security experts in academia and both the industry and the public sector.

[2]bot: Software that performs automated/repetitive tasks over the Internet.

Chapter 2

Extended background

2.1 Social Engineering (SE)

"Social Engineering uses influence and persuasion to deceive people by convincing them that the social engineer is someone he is not, or by manipulation. As a result, the social engineer is able to take advantage of people to obtain information with or without the use of technology." [Mitnick and Simon, 2002]

Social Engineering is the art of exploiting the weakest link of information security systems: the humans who are using them. A social engineer tries to manipulate her/his victims into divulging confidential information or performing her/his malicious objectives by using influence and persuasion. This form of attack renders technical shields ineffective as illustrated in Figure 2.1.

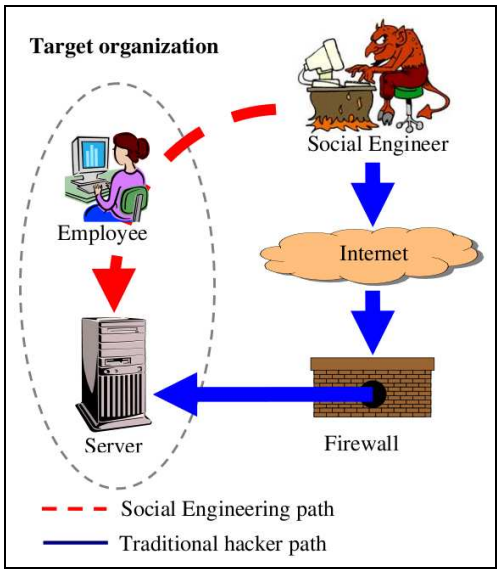

Figure 2.1. Social engineering approach [Hermansson and Ravne, 2005]

Furthermore individuals in general think that they are good at detecting these attacks. Research however indicates that humans perform poorly on detecting lies and persuasion [Grazioli, 2004, Qin and Burgoon, 2007, Marett et al., 2004].

The infamous attacks of Kevin Mitnick showed how devastating sophisticated social engineering attacks are for the information security of both companies and governmental organizations. While in the world of information and computer security, social engineering is most of the times studied by examples and stories (such as Mitnick's), the field of social psychology entails important findings on the principles of persuasion. Especially the work of Cialdini [2001], who is an expert in the field of persuasion is frequently cited within contributions to social engineering research. Although Cialdini exemplifies persuasion on the basis of marketing, his principles are crucial to understand how deception works. In the following the various facets of SE attacks are discussed and an introduction to Cialdini's six principles of influence is given. Understanding social engineering implies understanding common patterns of these attacks. Hence two different SE models are explained in section 2.1.3. Finally mitigation strategies for social engineering attacks are covered in section 2.1.4.

2.1.1 Various facets of SE attacks

Social engineering attacks are multifaceted and include: physical, social, and technical aspects which are used in different stages (more on this in section 2.1.3) of the actual attack. This subsection aims to explain the different approaches attackers use.

Physical approaches

With physical attacks, as the name implies, the attacker performs some form of physical action in order to gather information on a future victim which can range from personal information (social security number, date of birth) to valid credentials for a computer system. A common technique is searching through the trash of an organization, which is also know as *dumpster diving* [Granger, 2001]. An organization's dumpster can provide valuable information for attackers such as: personal data about employees, manuals, memos, and even print-outs of sensitive information such as credentials.
An attacker can also try to have a look at the office environment for information such as passwords written down on Post-it notes. Less sophisticated attacks of this kind involve *theft* or *extortion* to obtain information.

Social approaches

Obviously social attacks are the most emergent facet of social engineering which use socio-psychological techniques such as Cialdini's principles of persuasion (see section 2.1.2). According to Granger [2001] the most prevalent type of these social

attacks is conducted by phone. To increase the chances of these attacks, the perpetrators try to develop a relationship with her/his future victims beforehand.
Instead of contacting the victim, an attacker can try to make the victims ask for help from her/him. This indirect approach is known as *"reverse social engineering"* [Granger, 2001, Mitnick and Simon, 2002] and consists of three major parts: sabotage, advertising, and assisting [Nelson, 2008]. At first the attacker sabotages the company's computer system, this can reach from disconnecting someone from the company's network up to sophisticated manipulations of the victim's software applications. The attacker then advertises that she/he can fix the problem. Finally when the victim asks for help the social engineer resolves the problem she/he created earlier and while doing so asks the victims to comply with her/his requests (e.g. "Password is needed to fix the problem", "software installation required" etc.).

Technical approaches

Technical facets of attacks are mainly carried out over the Internet. Granger [2001] notes that the Internet is especially interesting for social engineers to harvest passwords, due to the fact that users often use the same (simple) passwords for different accounts. Furthermore most people are not aware that they give away a lot of personal information for free, which is useful to the attackers. Web search-engines are used by attackers to gather personal information about future victims. There are also tools available to gather and aggregate information from different web resources with Maltego[1] being one of the most popular. Social Networking Sites are becoming valuable sources for information as well, this issue is explained in more detail in section 2.2.1.

Combination of technical and social approaches

Successful social engineering attacks always use the different facets discussed so far in combination. Socio-technical approaches however have led to the most powerful weapons of social engineers. Examples include so called *baiting*, hereby attackers leave with malware infected storage media in a location where it is likely to be found by future victims. Those *"road apples"* could for example be USB sticks containing a Trojan horse [Stasiukonis, 2006]. Attackers are furthermore exploiting the curiosity of people by adding tempting labels to these road apples(storage media) e.g. "confidential" or "personal layoff 2009".

Another common combination of technical and social approaches is Phishing which Wikipedia [2009] defines as: *"Phishing is the criminally fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details by masquerading as a trustworthy entity in an electronic communication."* Phishing typically involves Email or instant messaging and in comparison

[1] Maltego is an open source intelligence and forensics application. It allows for the mining and gathering of information as well as the representation of this information in a meaningful way. http://www.paterva.com/maltego/

with social engineering aims, similar to Spam, at a large user group. Social engineering on the other hand is typically directed at single persons or small groups of people. Scammers hope that by the vast number of messages they send to users, enough people will get fooled and make their Phishing attack profitable. Herley and Florencio [2008] argue that the classical Phishing is not lucrative which can be understood as a reason why Phishing attacks are moving towards more sophisticated "spear" Phishing attacks. Spear Phishing attacks are highly targeted messages following initial data-mining. Jagatic et al. [2007] used social networking sites to mine data on students which then received a message that looked like being send from a friend of the victim. Jagatic et al. [2007] showed that they could increase the Phishing success rate from 16 per cent to 72 using this "social data". Hence spear Phishing can be seen as a marriage of social engineering and technology which is exactly what ASE is all about.

2.1.2 The six principles of Cialdini

Our inability to detecting social engineering attacks are caused by the fact that perpetrators according to Nohlberg [2008], make use of a weakness in the human psyche: so called "fixed-action patterns." This section discusses these "fixed-action patterns" on basis of Cialdini's six principles of influence. It should also be noted that according to Cialdini [2004], these principles in general are applicable in all cultures but cultural norms, traditions, and experiences weight the factors differently. There are other socio-psychological techniques, which are not covered in this thesis but are e.g. discussed by Nohlberg [2008], Levine [2003].

Reciprocation

The principle of reciprocation is firmly anchored within all human cultures. Reciprocation means that we should repay another person who does us a favor or gives something to us. This principle can be attributed to the human social revolution and meant you would do somebody a favor only if you are sure that you don't squander any of your own resources. Reciprocation is a very effective technique one comes across day-to-day, for example: product samples in supermarkets, favors of salesmen to add extras to an offer, free coffee for costumers etc. According to Cialdini [2001] even if the gift is very small, the request in return can be far greater because of one's urge to repay favors. A survey conducted during the Infosecurity Europe, supports this theory were 64 percent out of 300 office workers gave away their passwords for a chocolate bar [Register, 2008].

A variation of the Reciprocation principle is "Reject-then-retreat". A person is given two offers: an initial very demanding request; and a second lower and more sensible request. The first request will be rejected by most people but increases the success rate of the second retreating request.

Commitment and consistency

Like repaying a favor, being consistent is seen as an important quality in our society. The principle says that once somebody states her/his position on something, the person has to stick consistently with that commitment. As a tool of persuasion, this behavior is exploited by initially asking small favors in order to get a person's compliance with later requests.

Social Proof

People tend to mimic the behavior of others around them if they feel uncertain about something. We are likely to follow the behavior of other people because we assume if everybody is doing something, it has to be right. According to Cialdini [2001] our social proof behavior also leads to the ominous phenomenon of pluralistic ignorance where everybody waits to see the reaction of the other and finally nobody is doing anything at all. An attacker uses this fixed-action pattern to pursue victims by saying e.g. "everybody else in a company has complied with her/his request already" and therefore makes her/his malicious request harder to resist.

Liking and Similarity

People prefer to comply with requests of individuals they are in rapport with. The physical attractiveness of the requesting person can influence the likeness according to Cialdini [2001], this is also known as the "Halo-effect". Except from the attractiveness similarities are a useful tool to achieve rapport. Similarities can involve such things as common interests, same origin or background but also to have an enemy in common.

Authority

From our childhood on we are taught and raised to obey authorities as parents, teachers, police, as well as scientists and experts. People therefore tend to heed the advice and scrutinize requests less of those in a position of authority. Attackers also exploit the symbols which are commonly associated with authorities. This includes using fake titles or positions in combination with adequate cloth and trappings. A social engineering experiment at the United States Military Academy (USMA) at West Point showed that 80 per cent of all students fell for a fake request from an "Army Colonel" via Email [Ferguson, 2005]. One of the student's comment aptly summed up the effectiveness of this experiment on persuasion with use of authority: "The Email looked suspicious but it was from an Army colonel, so I figured it must be legitimate."

Scarcity

The scarcity principle states when an offer or the supply to a certain good is limited, we tend to give it a higher value. The information that others might be competing

for the same thing triggers our sense of competition. This can be observed in advertising everyday where terms such as "limited offer" are frequently used. Attackers exploit this behavior by e.g. telling victims "I'm very short of time but for you I would make an exception to change your computer's settings today."

2.1.3 SE models

Although the tools and techniques used by attackers vary with every social engineering attack, they follow a common pattern. Hence this section introduces two models to understand social engineering attacks: Mitnick's pioneer work with his *social engineering cycle*, and Nohlberg and Kowalski [2008] comprehensive *cycle of deception*.

Mitnick's SE cycle

Mitnick and Simon [2002] created a social engineering cycle (see Figure 2.2) in order to show common patterns of social engineering attacks. It is important to note that SE attacks always have a clear defined goal and that attackers iterate through the cycle's different stages until a specific (final) goal is reached. Gartner Inc. [2002a] described a similar cycle, with the main distinction being the different notions for the four stages.

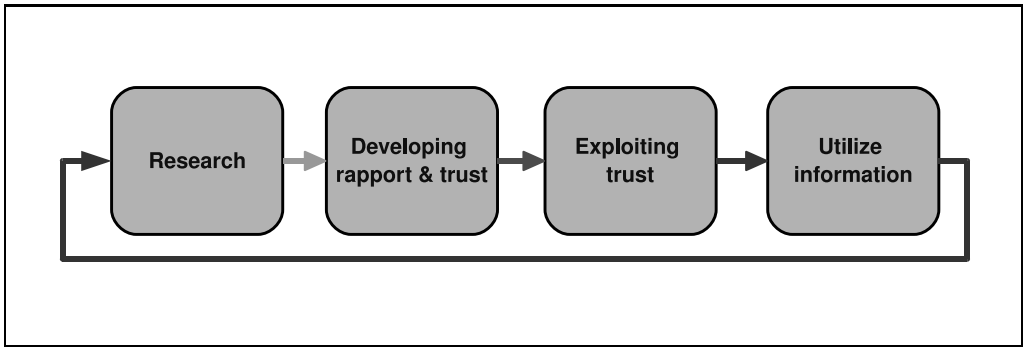

Figure 2.2. Social engineering cycle [Mitnick and Simon, 2002]

In the initial (*research*) stage the attacker gathers information on future victims through e.g. dumpster diving or web resources (see section 2.1.1). The attacker then aims to *develop rapport and trust* with her/his victims also using the knowledge from the *research* stage. Once a relationship is established, the attacker tries to *exploit* that trust by asking victims for a "favor" (e.g. credentials to a specific computer system). The attacker finally *utilizes* the gathered information to reach her/his final goal or returns to the first stage.

The cycle of deception

A more holistic model of social engineering attacks has been proposed by Nohlberg and Kowalski [2008]: "The cycle of deception" which includes not only an analysis from an attacker's standpoint but also from defenders and victims. Hence this model can be used to study attacks, to develop protection strategies and as a framework.

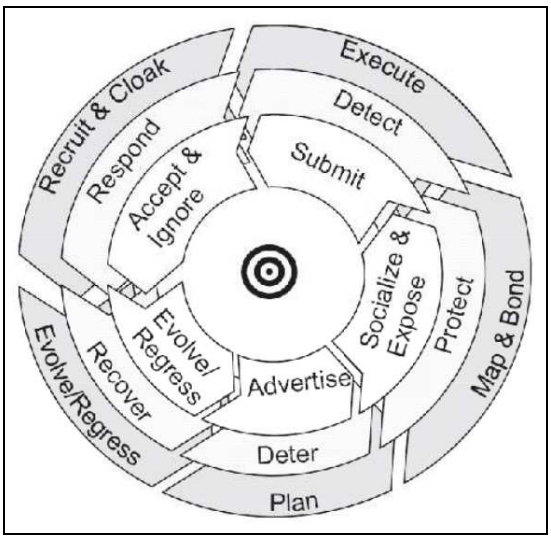

Figure 2.3. Cycle of deception [Nohlberg and Kowalski, 2008]

Attack cycle The outer cycle of the model represents the attack cycle which is initiated by the attacker creating a *plan* on how to reach his goal. The attacker then *maps* e.g. an organizations and *bonds* with future victims. Once the attacker is in rapport with her/his victims the *execution* of the actual attack follows e.g. asking for confidential information. After the execution the attacker tries to *cloak* the attack and maybe also to *recruit* the victim for future attacks. Finally the attacker can, depending on the success of the attack, *evolve* or *regress* such as to perform another simpler attack.

Defense cycle The middle cycle of the model represents the defenses between the victim and the attacker. By having a good public policy or a rumor of reporting incidents to the police, the defender can *deter* an attacker. By educating the employees about social engineering and having a strong policy on how to act, the defender *protects* the organization. By running for example a surveillance of the network communication, the defender can *detect* when sensitive data are sent. By enabling employees to report social engineering incidents in an easy and trustful way, the defender is able to *respond* to an ongoing attack. By knowing the value of information assets, having attacks reported and a well-designed information security policy, the defender can *recover* from the attack and learn from it.

Victim cycle The inner ring and most central part of the model represents the victim. By having something of value for attackers, the victim *advertises* its suitability as a target. By *socializing* with the criminal the victim sets itself up for deception, and by *exposing* personal information they make them accessible to the attacker. When the actual attack is executed, the victim *submits* to the attack, for instance by giving out the confidential information. After the attack has been executed the victim can choose to *accept* the attack by simply *ignoring* it or can *evolve* into someone who is harder to attack in the future. It is also possible that the victim can *regress*, turning into someone who accepts the role as a victim and is an easier target in the future.

2.1.4 Protection against SE

There is no silver bullet against social engineering although Mitnick and Simon [2002], Gartner Inc. [2002b] underline two important countermeasures: security policies and training of the personal. As social engineering attacks are carried out through different means, the policies need to cover a wide range of issues. Granger [2002] gives examples of social engineering risks and security policies to mitigate those risks, which the author called "combat strategies".

Area of Risk	**Hacker Tactic**	**Combat Strategy**
Phone (Help Desk)	Impersonation and persuasion	Train employees/help desk to never give out passwords or other confidential info by phone
Office	Shoulder surfing	Don't type in passwords with anyone else present (or if you must, do it quickly)
Phone (Help Desk)	Impersonation on help desk calls	All employees should be assigned a PIN specific to help desk support
Office	Wandering through halls looking for open offices	Require all guests to be escorted
Dumpsters	Dumpster diving	Keep all trash in secured, monitored areas, shred important data, erase magnetic media
Intranet-Internet	Creation and insertion of mock software on intranet or internet to snarf passwords	Continual awareness of system and network changes, training on password use
Office	Stealing sensitive documents	Mark documents as confidential and require those documents to be locked
General-Psychological	Impersonation and persuasion	Keep employees on their toes through continued awareness and training programs

Table 2.1. Common intrusion tactics and strategies for prevention [Granger, 2002]

Gragg [2003] points out that any education on social engineering must include psychology and persuasion in order to understand and counter attacks. Mitnick and Simon [2002] advise training on the security policies and procedures but also awareness training. An interesting approach on how to raise the awareness for social engineering attacks and train employees is the work by Hermansson and

Ravne [2005] on fighting social engineering by combining scenario based learning with the psychological factors of persuasion. Yet another approach by Srikwan [2008] recommends cartoons to teach users about social engineering and Phishing.

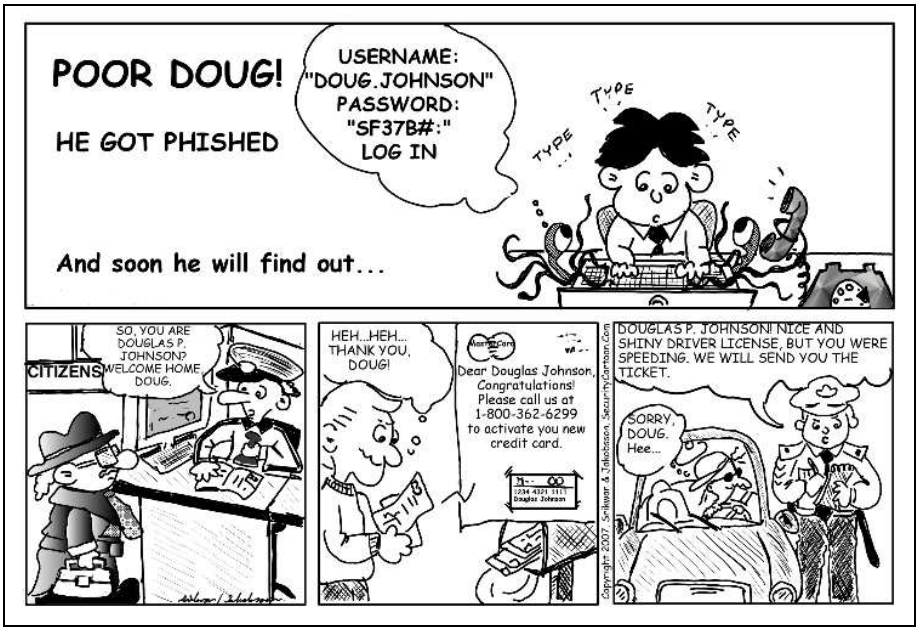

Figure 2.4. Cartoon on phishing with keyloggers [Srikwan, 2008]

In conclusion it is important to note that the key to counter social engineering is of course the personal. Security education is its own research field and the best approach to train personal also depends on the organization. Large organizations might find "A Multi-Level Defense Against Social Engineering" by Gragg [2003] useful, while other organizations might use game-based learning [Näckros, 2007] or even train their personal with chatbots. A starting point for the defense against social engineering is the cycle of deception's defense cycle from Nohlberg and Kowalski [2008].

2.2 Social Networking Sites (SNSs)

"A social network service focuses on building online communities of people who share interests and activities, or who are interested in exploring the interests and activities of others." [Wikipedia, 2008a]

Social network services in general are classified into *internal* and *external* services, whereas the first type is used in closed environments with restricted membership, while in the second case an external service is consumed. The term "social network service" does not constraint which type of software client is offered and used to access the service. Social Networking Sites (SNSs) are a specific type of external social network service and are solely web based. Hence SNSs are accessed via web browser over the Internet.

Figure 2.5. Example SNS profile (the author's profile on Facebook)

At the present time SNSs are one of the fastest growing web application subgroup which are often marketed as being part of the so called "Web 2.0". Finding a widely accepted definition for Web 2.0, is a futile effort as the term is meant to discuss different aspects of successful online services in the post dot-com era. One view on Web 2.0 applications amongst many is the discussion around user-generated data and information which is seen as one of the major success factors of Web 2.0 applications [O'Reilly, 2008]. The most famous example for user generated content is Wikipedia, which is an online lexica entirely based on articles contributed by its

users. The user generated data within SNSs are personal information and different forms of media created by its users.

Two of the by far most popular SNSs are *MySpace* and *Facebook*, whose userbase is constantly growing. Between June 2006 and June 2007 Myspace was able to increase by 72 per cent. Facebook had an even more dramatic increase by 242 per cent of unique visitors in the USA [Comscore, 2008]. The table below (Table 2.2) gives an overview on today's largest SNSs based on the number of registered users. It should be noted that a ranking on basis of the actual number of active users would be more significant, but no reliable resources exist for these values.

Name	**Description/Focus**	**Registered users**
MySpace http://myspace.com	General. Popular in the United States, Canada and Europe.	246,351,193
Facebook http://facebook.com	General. Started as a network for college students, now open to everyone.	124,000,000
MS. Live Spaces http://spaces.live.com/	Blogging (formerly MSN Spaces)	120,000,000
Habbo http://habbo.com	General. Over 31 communities worldwide. Chat Room and user profiles.	100,000,000
hi5 http://hi5.com	General. Popular in Portugal, Cyprus, Romania, Thailand and Latin America.	80,000,000
Friendster http://www.friendster.com	General. Popular in Southeast Asian countries.	80,000,000
orkut http://www.orkut.com	General. Popular in Brazil, Paraguay, India and Pakistan.	67,000,000
Flixster http://www.flixster.com	Movies	63,000,000
Classmates.com http://www.classmates.com	School, college, work	50,000,000
Reunion.com http://www.reunion.com	Locating friends and family, keeping in touch	48,000,000

Table 2.2. SNSs Top 10 by number of registered users [Wikipedia, 2008b]

Table 2.2 illustrates that the popularity of SSNs varies within different countries e.g. MySpace is most popular in North America, mixi (http://mixi.jp) the most popular SNS in Japan, and orkut the most popular in Latin America. Except from their different geographical spread, SNSs also focus on different target groups e.g. LinkedIn and XING on business, GENi on genealogy, and Classmates.com on schools/colleges. Although SNSs focus on different target groups and countries, the main functionalities SNSs provide to their users are nevertheless the same:

- Creation & maintenance of a personal online profile
- Cultivation of social relationships with other users
- Possibilities to share media with other users
- Management of permission rights to personal information and media

2.2.1 Security risks of SNSs usage

The majority of scientific publications on information security threats related to SSNs usage are about the implications of these applications for privacy. Gross and Acquisti [2005] analyzed the online behavior of 4,000 Carnegie Mellon University students and concluded that the students have not been aware of the ways their personal information could be exploited. Online disclosure of personal information from SNSs is furthermore discussed by Rosenblum [2007], Kolek and Saunders [2008], Gibson [2007]. These articles discuss the security issues on basis of privacy risks. The ENISA[2] position paper [Hogben, 2007] on the other hand introduces four threat categories, which are more practical to understand all the information security risks that are involved with SNSs usage. Hence the following sections discusses the security threats of SNSs usage with Hogben [2007].

Privacy related threats

i **Digital dossier aggregation** SNS profiles can be fetched and stored by third parties in order to create a digital dossier of personal data. Hogben [2007] argues that due to diminished costs of disk storage and Internet downloads it is feasible to take incremental snapshots of entire SNSs. A proof-of-concept digital dossier aggregation, carried out on an early version of the most popular German SNS (studiVZ), showed that 1.074.574 profiles could be aggregated within less than four hours with a computer cluster consisting of ten computers [Fritsch, 2008].

ii **Secondary data collection vulnerabilities** SNS members also disclose information to their internet service providers (ISPs). While this is not solely limited to SNSs, the main difference is the extent of coherent personal data exposed to ISPs. For example to map the circle of friends without SNSs data, ISPs

[2]European Network and Information Security Agency

need to correlate information from multiple Email addresses, instant messaging, etc. Even more important is the threat of disclosure and resale of personal information to third parties for example to providers of targeted advertisement. The section on the economics of SNSs (see section 2.2.2) further discusses the commercialization of personal information.

iii **Face recognition vulnerabilities** SNS users provide profile images of themselves and SNSs contain shared images associated with them. Face recognition technology can be used to identify users across different SNSs no matter if pseudonyms or fake names are being used.

iv **CBIR (Content-based Image Retrieval)** CBIR is a technology which deduces the location of users by analyzing and comparing common patterns in images. Hence shared images within SNSs not only disclose the identity of users but possible the location of users as well.

v **Linkability from Image Metadata, Tagging and Cross-profile Images** While users control which information and media they share within a SNS, they can't control which content other users upload and link to their profile. Images might also contain metadata including the serial number of the camera used to make the pictures.

vi **Difficulty of Complete Account Deletion** Users that wish to deactivate their SNS account in most cases face difficulties to do so. Partly because not all comments and messages sent to other users will be deleted and also because SNS providers keep backups of account data.

SNS variants of traditional network and information security threats

i **Social Networking Spam** As SNSs steadily grow, they have become interesting targets for spammers. The use of SNS spamming software furthermore automates the process of sending unsolicited bulk messages. The Spam content can reach from advertising to Phishing messages. A study based on anonymized headers of 362 million messages exchanged by 4.2 million users of Facebook, claimed that 43 per cent of all messages analyzed were to be considered as Spam [Golder et al., 2006].

ii **Cross Site Scripting, Viruses and Worms** In order that users are able to customize the design of their profiles, SNSs often provide the possibility to post HTML code. Furthermore third party applications (widgets) are used to extend the functionality of SNSs and together with HTML code they state a risk for XSS[3] vulnerabilities. *Samy/JS.Spacehero* for example was a XSS worm

[3]XSS: "Cross-site scripting(XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. Examples of such code include HTML code and client-side scripts."
[Wikipedia, 2008c]

on MySpace, which infected more than one million profiles within the first 24 hours.

iii **SNS Aggregators** Social Aggregators offer services to integrate the data from different web services and SNSs into a single platform. Popular services include Gathera, FriendFeed, Spokeo and Secondbrain. As with all single-sign-on systems, the access to multiple services (in this case SNSs) depends on only one password which if selected badly states a single point failure. These services are also used to correlate user data across different SNSs. Spokeo for example provides a charged service which aggregates data of 41 social networks with someone's Email address being the only information required [Spokeo, 2008].

Identity related threats

i **Spear Phishing using SNSs and SN-specific Phishing** Spear Phishing attacks are targeted Phishing attacks. The information available through SNSs is harvested by scammers and used as a basis for a spear Phishing attack. SNSs are furthermore used as a medium for carrying out the Phishing attack itself rather than using standard Email messages.

ii **Infiltration of Networks Leading to Information Leakage** SNSs allow users to define who has access to their personal information for example by giving access to certain "friends" or by defining restricted groups (networks). These are important features to improve the privacy issues of SNSs usage but once a closed network is infiltrated the protection is rendered useless.

iii **Profile-squatting and Reputation Slander through ID Theft** Profile-squatting is similar to domain-squatting only that instead of Internet domains persons are targeted. Fake profiles are set-up in the name of someone else in order to slander her/his reputation within a certain network. Examples include the Moroccan computer engineer who set up a name of a member of the royal family [BBC News, 2008].

Social threats

i **Stalking** According to Wikipedia [2008d] stalking is *"a perjorative term for the obsessive following, observing, or contacting of another person or the obsessive attempt to engage in any of these activities"*. SNSs can be misused by perpetrators to contact their victims but also to gather information on them. SNSs users often disclose location data via their pictures (see CBR) or personal information.

ii **Cyber-bullying and grooming** Cyber-buylling are aggressive attacks and bullying attempts carried out over the www, while cyber-grooming refers to attempts by adults to approach minors via the web to abuse them sexually. One of the most infamous cases involving cyber-bullying, the "Megan Meier

case", led to the suicide of a teenage girl [Suburban Journals, 2008]. In the Meg Meier case the perpetrator exploited the ease of setting up a fake profile, which was also used in a recent cyber-grooming case [Tribune, 2008].

iii **Corporate Espionage** Hogben [2007] discusses the threat of corporate espionage via SNSs regarding social engineering. A social engineering can use SNSs to collect valuable information about employees (employees of a specific organization, position within the organization, full name, Email addresses etc.) rather than infiltrating an organization and using classic social engineering approaches.

2.2.2 The economics of SNSs

Social networking sites follow to some degree the same business paradigm as all Web 2.0 services which means that these services are free of charge and make profit by selling online advertising services to third parties. Hence the number of users and signups are critical for the commercial success of SNSs. SNS providers therefore design their services in order to increase the number of new signups and target broader user scopes. Facebook, for example, was initially only accessible to students of Harvard University and was rapidly expanded to more colleges and universities. Since September 2006, everyone in possession of an Email address can join Facebook, whereas before the service was limited to work, university, and alumni Email addresses [Facebook, 2006]. Even more important than the number of visitors these websites attract, is the socio-demographic datapool created by their users. This means that advertisers can target a certain user pool (e.g. "All married, Swedish men in the age of 30 to 55 years.") and are not solely dependent on contextual advertising such as Google Ad Sense, Yahoo! Publisher Network or Microsoft adCenter. The expectations in targeted advertising through SNSs was reflected in the purchase of MySpace for 580m US$ by the News Corp. in 2005 [BBC News, 2005]. By this time MySpace claimed to have 22 million registered users [BusinessWeek, 2007] which means News Corp. paid roughly 30 US$ per user profile. In October 2007, Microsoft bought a 1.6 per cent share of Facebook for 246m US$ [Microsoft, 2007]. This implied a fictional value of 15.375b US$ for Facebook or 220 US$ for every of the 70m user profiles at this time [CNET News, 2008].

Fogg and Iizawa [2008] compared how SNS in different cultures motivate users toward sharing personal information by using persuasion tactics. According to Fogg and Iizawa [2008] *"the Facebook service is designed to persuade users to take quick action with pointed outcomes"* while the Japanese SNS Mixi uses a more subtle approach. The research makes clear that persuasion of SNSs users to disclose more personal information is an essential SNS feature even though there are cultural differences on how the SNS is designed persuasively. Persuasive features of SNSs to disclose personal information help the SNSs providers to create a more valuable data pool and thus generate more profit.

2.3 Proof of concept ASE bot: tools and methods

2.3.1 Web scraping

Web scraping is a technique to extract data from websites. While websites are created to be readable by humans they are not necessarily readable by machines. Web scraping therefore aims at transforming unstructured HTML data into structured data that can be stored and processed by other applications. The simplest form of web scraping is using *copy & paste* to extract certain information from web pages. Manual approaches are time-consuming and special web scraping software therefore automates the data extraction process. This is achieved by software that mimics a web browser which is also used for web indexing with so called "web bots" or "spiders".

Traditional tools and approaches

LibWWW is an early web browser which is solely text based and which was developed for the use on terminals. Since the initial implementation of LibWWW[4] in 1991, the library has been implemented in different script languages (e.g. Perl) and is also used in modern text based browsers such as lynx[5]. The inclusion of the library in programming and script languages made it a favorable library to create custom www clients. For example Jagatic et al. [2007], Acquisti and Gross [2006] used Perl's LWP library to web scrape SNSs and harvest information for their experiments.

Tag soup parser While LibWWW offers a convenient interface to the HTTP protocol the required information still needs to be extracted from the fetched web pages. Regular expressions are a powerful tool and one way to achieve the data extraction but require some level of expertise. Even more importantly, regular expressions require that the webpage is following the HTML syntax in a consistant way. Web developers, however, are often not following the web standards strictly as HTML browsers treat syntax errors leniently. Web pages that follow the W3C standards to some degree and include design and programming errors are called *"tag soups"*. A number of libraries exist which help developers to parse these tag soups and enable data extraction:

- Beautiful Soup (Python)
  Beautiful Soup is a Python HTML/XML parser designed for quick turnaround projects like screen-scraping.
  http://www.crummy.com/software/BeautifulSoup/

[4] http://www.w3.org/LineMode/
[5] http://lynx.isc.org/

- Hpricot (Ruby)
  Hpricot is a nice, loose HTML parser for Ruby, written in C.
  http://github.com/why/hpricot/tree/master

- TagSoup (Java)
  TagSoup is designed for people who have to process this stuff using some semblance of a rational application design.
  http://home.ccil.org/ cowan/XML/tagsoup/

Web automation Web scraping often requires interaction with web pages which means the software should automate the browsing through a webpage. For example it should perform the following sequence: load "http://dsv.su.se", click "Internt", open the link to "Daisy", log-into "Daisy" and fetch all calendar entries. This functionality could in theory be implemented with the earlier described LibWWW library but web automation libraries have two big advantages: they help to automate interactions with websites by providing abstractions of common web browsing tasks and they support advanced browser features (cookies, frames, etc.). Two popular tools are HttpUnit and Mechanize:

- HttpUnit (Java)
  Written in Java, HttpUnit emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages either as text, an XML DOM, or containers of forms, tables, and links.
  http://httpunit.sourceforge.net/

- WWW::Mechanize (Perl, Ruby, Python)
  Mechanize or Mech for short, helps to automate interaction with a website.
  http://search.cpan.org/dist/WWW-Mechanize/ (Perl)
  http://mechanize.rubyforge.org/ (Ruby)
  http://wwwsearch.sourceforge.net/mechanize/ (Python)

Web automation & tag soup parsing in combination Common web scraping applications use web automation in combination with tag soup parsing. The following Ruby script created by Brook [2009] is an example of how to use web scraping to get a list of unread Email from Gmail:

```
require 'rubygems'
require 'mechanize'

agent = WWW::Mechanize.new

page = agent.get 'http://www.gmail.com'
form = page.forms.first
form.Email = '***your gmail account***'
```

```
form.Passwd = '***your password***'
page = agent.submit form

page = agent.get page.search("//meta").first.attributes['href'].gsub(/'/,'')
page = agent.get page.uri.to_s.sub(/\?.*$/, "?ui=html&zy=n")
page.search("//tr[@bgcolor='#ffffff']") do |row|
  from, subject = *row.search("//b/text()")
  url = page.uri.to_s.sub(/ui.*$/, row.search("//a").first.attributes["href"])
  puts "From: #{from}\nSubject: #{subject}\nLink: #{url}\n\n"
  email = agent.get url
end
```

The above script illustrates how pragmatic and straight forward web scraping can be achieved in this case with a combination of Mechanize and Hpricot with the Ruby scripting language. There are however two main reasons why combining web automation and tag soup parsing might prove unsuccessful, namely the *User agent string* and *JavaScript* support:

User agent string (UAS) Every WWW client submits information about itself (application name & version and the used operating system) to the web server it accesses for example:

- "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1;)"
  UAS from Internet Explorer 6.x on Microsoft Windows XP.
- "Mozilla/5.0 (compatible; Googlebot/2.1; http://google.com/bot.html)"
  UAS from Googlebot which is used by Google Inc. to index web sites.
- "Python-urllib/2.5"
  UAS of Python Mechanize package.

Web site providers can detect and than block certain user agent strings. Programmers of web scraping applications may however still access these web sites by spoofing the user agent strings of their www clients.

JavaScript support State of the art web automation libraries lack support of JavaScript. WWW:Mechanize doesn't support JavaScript at all [CPAN, 2009] while HttpUnit implements merely a basic subset of the JavaScript 1.1 functionalities [Gold, 2009]. Today's web sites make, however, extensive use of JavaScript also because of AJAX[6] which plays an important role in Web 2.0 internet applications such as Google Docs or Yahoo's Flickr.

[6] Asynchronous JavaScript and XML

Scripting with web browser extensions

A different approach to web scraping than using a self-programmed www client is to "script" existing Web browsers. This means embedding the web automation tool inside a standard browser, the content of a web page can be accessed as the user sees it [Bolin et al., 2005]. LiveAgent [Krulwich, 1997] and WebCVR [Anupam et al., 2000] are two early implementation that could be used to record browser actions and replay those actions at a later point. In this section the focus lies on tools which work with the two most popular web browsers [Net Applications, 2009] at time of writing: Microsoft's Internet Explorer and Mozilla's Firefox.

Chickenfoot Chickenfoot[7] is an extension for the Mozilla Firefox web browser that is being developed by the User Interface Design Group of MIT. The chickenfoot extension itself is integrated as a Sidebar into Firefox and offers debugging and recording functionalities.

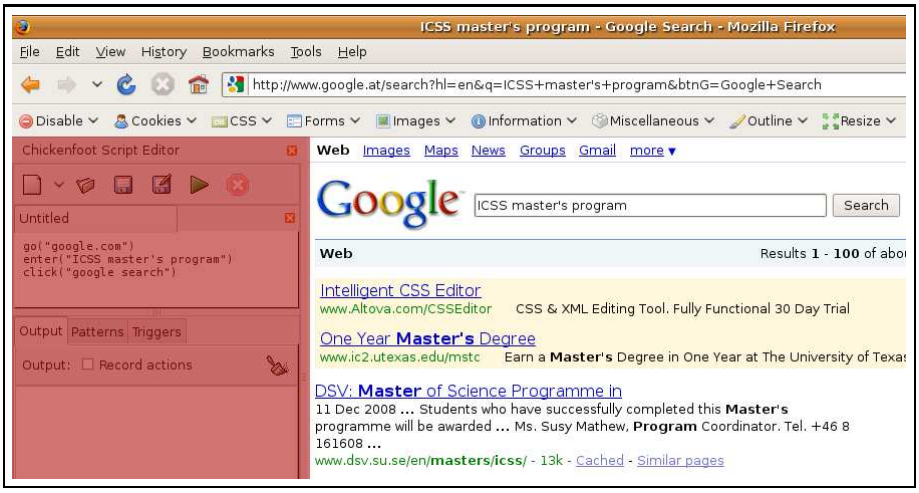

Figure 2.6. Sidebar of the chickenfoot extension in Mozilla Firefox

Chickenfoot offers a programming environment on top of Firefox that can be used for both web automation and the manipulation of web sites. Chickenfoot scripts are written as a superset of JavaScript functions in its own language called "Chickenscratch". The primary target group of this extension are end-users. The developers of chickenfoot therefore implemented a heuristic keyword matching algorithm to help users identify certain components of a web site without requiring them to have deep understanding of a web page's underlying structure [Bolin et al., 2005]. For example, this simple script searches for the ICSS master's program in Google:

```
go("google.com")
```

[7] http://groups.csail.mit.edu/uid/chickenfoot/

```
enter("ICSS master's program")
click("google search")
```

As JavaScript can be mixed and combined with chickenscratch scripts the extension is also interesting for developers. Chickenfoot for example doesn't have a "dump" command which is needed for web scraping, using JavaScript however this can be implemented easily[8].

IMacros IMacros[9] is a web browser extension for Internet Explorer and Mozilla Firefox. Two versions of IMacros are available: a free and a proprietary one. The free version only supports web automation while the commercial version includes image recognition, file upload functionality, a command line interface and web scraping functionality [iOpus Inc., 2009].

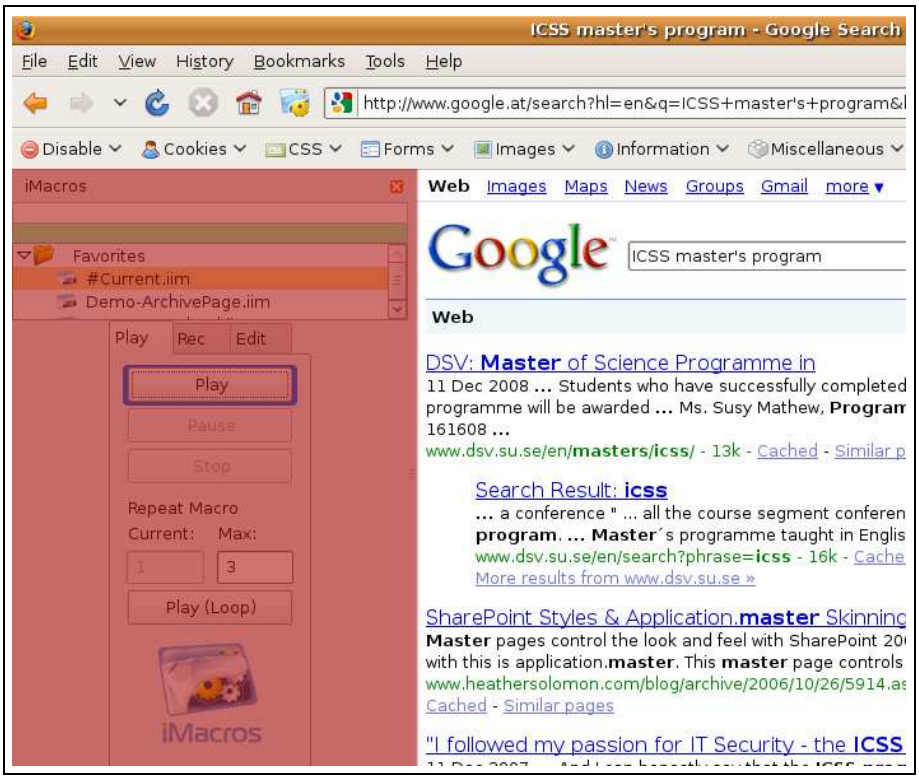

Figure 2.7. Sidebar of the IMacros extension (free version) in Mozilla Firefox

As compared with Chickenfoot, IMacros requires more expertise on the programming level as the syntax is more complex but at least in the commercial version it offers a larger feature set. Another function which IMacros enables, is the possi-

[8]http://groups.csail.mit.edu/uid/chickenfoot/scripts/index.php/Dump_current_HTML
[9]http://www.iopus.com/imacros

bility for users to share scripts amongst each other. For example searching for the ICSS master's program in the IMacros syntax looks like:

```
URL GOTO=www.google.com
TAG POS=1 TYPE=INPUT:TEXT FORM=NAME:f ATTR=NAME:q CONTENT=ICSS<SP>master's<SP>program
TAG POS=1 TYPE=INPUT:SUBMIT FORM=NAME:f ATTR=NAME:btnG\&\&VALUE:Google<SP>Search
```

Piggy Bank Piggy Bank[10] is an extension that works with Mozilla's Firefox and the Flock browser. The extension was designed to help users with utilizing semantic information provided by websites and re-combine this information. As this semantic information is only provided by few websites today [Huynh et al., 2007], Piggy Bank includes web scraping functionality that helps to transform tag soups into RDF data (more on RDF in section 2.3.2). The developers of Piggy Bank further created a web server application for sharing semantic web information which has been gathered by users. This online service should enable collaborative efforts to build a sophisticated semantic web information pool with the help of Piggy Bank users.

2.3.2 W3C Ressource Description Framework (RDF)

One of the objectives the World Wide Web Consortium(W3C) is continuously persuiting, since its foundation in 1994[11], is the transformation of the world wide web (www), as we know it today, towards the 'semantic web'. The semantic web could be described in a few words: making information (within the www) machine-accessible. This statement might sound odd first as all information available trough the www is already digital and this issue becomes more obvious if one takes a look at today's search engines. As websites contain either no or poor metadata, websites are analyzed with the help of metrics (number of words, structure, keywords etc.) and therefore it is impossible to interpretate the published information. The example of Antoniou and Van Harmelen [2004] illustrates that it is difficult for the current search engines to distinguish the meaning of:

I am a professor of computer science. from
I am a professor of computer science, you may think. Well, ...

Therefore the W3C introduced the resource description framework (RDF), which uses triples of knowledge to define meta data about web ressources. Each triple consists of a subject, a predicate, and an object. RDF information is thus expressed as a list of triples.

[10] http://simile.mit.edu/wiki/Piggy_Bank
[11] http://www.w3.org/Consortium/

The previous example would then be represented like:

SUBJECT	PREDICATE	OBJECT
I	am_a	computer science professor

Table 2.3. An example RDF triple

This triple seems reasonable and to make it certainly machine-readable, RDF defines the following standards.

RDF/XML A XML syntax for RDF. The above example would look like:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
 xmlns:foaf='http://xmlns.com/foaf/0.1/'
 xmlns:myrdfs='http://example.com/myrdfs/0/'
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
>
 <foaf:Person>
  <foaf:name>I</foaf:name>
  <myrdfs:am_a>professor of computer science</myrdfs:am_a>
 </foaf:Person>
</rdf:RDF>
```

RDF schema (RDFS) Defining vocabulary for RDF. This is important for both applications to 'understand' the RDF-XML triples as well as for design guidelines for RDF developers. In the XML example above two RDF schemes are used, FOAF [12] and a fictitious schema named 'myrdfs'.

Simple protocol and RDF query language (SPARQL) The recommended language to be used to query RDF triples.

RDF implements a first-order predicate calculus (FOPC) which means that knowledge can be deduced from RDF triples of knowledge. A classic example for a deduction are the two statements: "All men are mortal" and "Socrates is a man". FOPC can deduce the answer to the question "Is Socrates mortal?".

In order to fully understand the W3C-RDF standard, I recommend the RDF Primer [Manola et al., 2006] which is a helpful ressource with continuative information.

[12] Friend of a Friend RDF-schema: http://xmlns.com/foaf/spec/

2.3.3 Artificial Conversational Entity (ACE)

ACEs are also known as chatbots (or chatterbots) and are computer programs which simulate intelligent conversations through text or speech. The initial idea of such a computer program is attributed to the computer science pioneer Alan Turing who stated the philosophical question: "Can machines think?" [Turing, 1950]. He furthermore proposed a method to answer this question: "the imitation game". In this game a third person (interrogator) is talking with two other persons and has to decide which one is a real person and which a computer program. Turing [1950] gave an example what a conversation with a computer program should look like:

```
Q: Please write me a sonnet on the subject of the Forth Bridge.
A : Count me out on this one. I never could write poetry.
Q: Add 34957 to 70764.
A: (Pause about 30 seconds and then give as answer) 105621.
Q: Do you play chess?
A: Yes.
Q: I have K at my K1, and no other pieces. You have only K at K6 and R at R1.
   It is your move. What do you play?
A: (After a pause of 15 seconds) R-R8 mate.
```

Two early implementations of these ACEs proposed by Turing [1950] have been ELIZA [Weizenbaum, 1966] and PARRY [Enea et al., 1973]. In 1990 an annual competition for artificial intelligence was started which is based on the "Turing test" (the imitation game). At the moment chatterbots are being used primary in commercial environments (e.g. IKEA's virtual assistance "Anna"[13]) and in education (e.g. in security education [Kowalski et al., 2008]).

Artificial Linguistic Internet Computer Entity (ALICE)

ALICE is a state-of-the-art chatterbot created by Richard S. Wallace which has won the Loebner competition in 2000, 2001 and 2004 [Loebner, 2009]. The original program was created in 1995 and uses the Artificial Intelligence Markup Language (AIML) as a basis [Wallace, 2000]. The specifications of ALICE/AIML have been made public and licensed under the GNU General Public License. Since its initial release, ALICE has been developed further and implementations of AIML interpreters have been written for different programming languages such as Java, C++, Lisp, PHP, and Perl. Because ALICE/AIML is open source it also forms the basis of many commercial and non-commercial chatterbots (a list can be found online at: http://www.alicebot.org/aimlbots.html).

[13] http://www.ikea.com/us/en/

AIML (Artificial Intelligence Markup Language)

AIML is an XML dialect which is used to program the logic of ALICE. The most important AIML XML tags are:

AIML tag	description
<aiml>	denotes the beginning and end of an AIML document
<category>	denotes a "unit of knowledge" of ALICE knowledge base
<pattern>	denotes a simple pattern that is used to match user input
<template>	denotes responses to user input

Table 2.4. Most important AIML tags

The following AIML examples shows how ALICE can be programmed to reply to a simple question.

```
<aiml>
<category>
  <pattern>HOW ARE YOU CALLED?</pattern>
  <template>My name is <bot name="name"/>.</template>
</category>
</aiml>
```

Although AIML is an easy syntax to learn and simple to use, it takes a considerable amount of time to program a functioning chatterbot. Open source AIML sets are available online for free[14] and include ready to use French, German, Italian and additional ALICE AIML sets. These freely available AIML sets are a good starting point for programming an AIML chatterbot.

[14] http://aitools.org/Free_AIML_sets

Part II

Automated Social Engineering

Chapter 3

A holistic view on ASE via Facebook

"Much like the current botnets, which are threatening due to their size and network communicating possibilities, the automated social engineering bots will be threatening due to the fact that they will know so many people and so much about them. Their threat will lie in abusing the social networks, and not the Internet traffic networks." [Nohlberg et al., 2008]

Social engineering always starts with gathering background information on possible future victims, for example through dumpster diving, phone calls or similar approaches (see section 2.1.1). Because of the emerging usage of SNSs the toolset available to attackers changes, as they can now use SNSs such as Facebook to gather the initial background information on future victims. Furthermore as all data in SNSs is available in digital form and SNSs include tools to communicate the whole social engineering attack can be automated. Special software, so called Internet bots, is already being used to spread spam messages in SNSs. In a recent case with Facebook a malicious attacker first stole Facebook credentials via Phishing and then used these accounts to send more then four million unsolicited bulk messages [SPAMfighter, 2008]. More sophisticated malicious bots, which make use of a chatbots to fool their victims, have already been seen in the wild [Epstein, 2007, The Guardian, 2007]. Automated social engineering might be initially misunderstood as a change in the toolset of a social engineer but this attack has profound implications. The goal of automation is to reduce the human intervention time to a minimum. Classic social engineering attacks are expensive due to the fact that it is time intensive to build rapport with someone, maintain, and finally exploit the relationship. Automated social engineering bots on the other hand require little human time, are scalable and therefore make social engineering a cheap and attractive attack. We chose to analyze ASE in depth on basis of the social networking site Facebook and furthermore implement the proof of concept ASE bot for the Facebook web service. Facebook is an attractive platform for ASE attacks because of its huge number of members and their socio-demographic characteristics; at the time of writing Facebook [2009b] claims to have 175 million active users.

3.1 A holistic analysis of ASE attacks on Facebook

We used the content/subject areas of the Systemic-Holistic approach by Yngström [1996] which includes technical as well as non-technical aspects to show the big picture of possible ASE attacks carried out on basis of Facebook's social networking platform. Our analysis includes the technical, operational, administrative and managerial, and legal aspects of Facebook that we found to be most relevant for automated social engineering. While we use these four subject areas to shed more light on the protective measures of Facebook we use the area on ethics to discuss research on ASE (see section 3.2). In section 3.3 we tackle these issues from the organizational perspective in order to better understand why organizations that are using Facebook are at risk for ASE attacks.

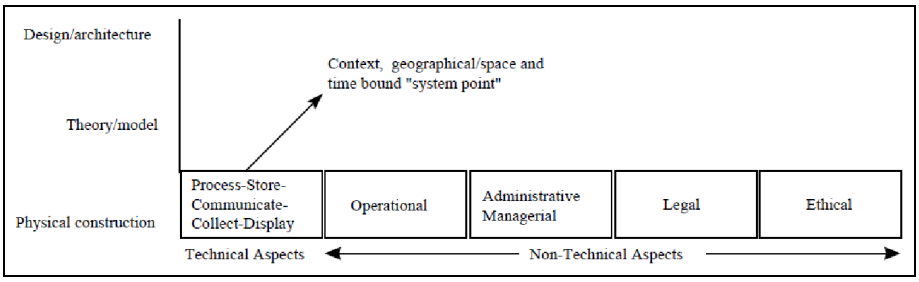

Figure 3.1. Systemic-Holistic Approach to IT Security [Yngström, 1996]

3.1.1 Technical aspects

Authentication

Email addresses in combination with passwords are used for authentication in Facebook. New users sign-up with their Email address and then get a confirmation link sent to the Email address they specified. The confirmation link ensures that the Email address is valid and in possession of the person signing up. Because Facebook doesn't restrict the type of Email address to work and educational addresses anymore, anybody in possession of a valid Email address can sign-up to Facebook [Facebook, 2006]. Email addresses are furthermore the authentication factor to prove eligibility for network memberships. Facebook is made up of networks whereas the membership to regional networks (e.g. "Sweden", "Stockholm") is open and membership to workplace, and high school/college networks is in general closed. In order to join a closed network users have to possess a valid Email address for the specific network, e.g. to join the Facebook network of "Stockholms universitet" users need to have a valid Email address from Stockholms universitet ending in "su.se". With this Email-based authentication mechanism Facebook protects against users making false claims about their work and education affiliation.

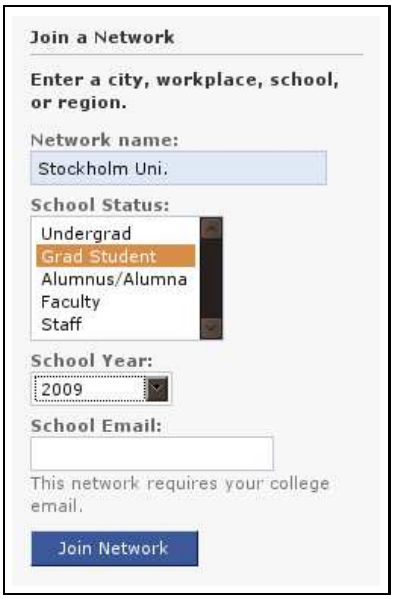

Figure 3.2. Joining a closed network on Facebook

Authenticated state

To maintain the authenticated state after a successful login Facebook uses HTTP cookies and hidden form fields. The cookies contain session information while the hidden form fields ensure that forms are submitted from users and protect against CSRF[1] attacks. Because the hidden form values are partly created by JavaScript this authentication method also blocks bots that are not JavaScript compatible.

Applications

Facebook has already suffered from common web applications vulnerabilities such as cross side scripting in the past [Pagkalos, 2008]. What makes Facebook different from common web sites is the fact that Facebook offers an API for developers to create their own applications. These third party applications are an additional attack vector which could be exploited by an ASE bot either for malicious actions (e.g. distributed denial of service attack [Athanasopoulos et al., 2008]) or to gather personal information on future victims.

captcha

CAPTCHA stands for "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part". Captchas are challenges that should be easy for humans but difficult for computers to solve. Facebook uses image-based captchas where distorted text has to be read and replied back as a challenge response. Certain actions

[1] Cross-site request forgery

in Facebook such as sign-up or adding friends are protected with a captcha to block automated programs such as spam bots.

Figure 3.3. Example of a captcha on Facebook to protect the sign-up process

These captchas could in theory be broken with optical character recognition (OCR) algorithms but this requires a high level of sophistication. Captchas are however not resistant to human attacks, Robertson [2007] for example reported one case where scammers used visitors of a striptease website to solve captchas for them.

3.1.2 Operational aspects

Automatic pattern matching & security metrics

Facebook has implemented automated mechanisms to detect abusive behavior. These mechanisms are however not communicated in a transparent way. Once a possible abuse of a Facebook feature is detected users are first warned and if they don't adapt their using habits their accounts get permanently disabled. Facebook intentionally doesn't disclose any details on why certain users have been warned or their accounts disabled, for example Facebook's explanation for a usage warning is:

"Facebook has determined that you were using a feature at a rate that is likely to be abusive. For example, if you were warned about approaching the limit for sending friend requests, then we determined that you were adding new friends too quickly. Please be aware that further abuse of such features can result in your account being permanently disabled." [Facebook, 2009a]

There have been numerous blog posts from people whose Facebook accounts have been disabled. Muller [2008] aggregated this information to point out possible reasons why people get their Facebook accounts blocked. In the following we give an estimated list of security metrics that might detect an ASE bot:

i *Fake names in profile*
For example nicknames or only initials, a name such as "ASE bot" would probably get detected as well.

ii *Exceeded rate limits*

Rate of joined groups.

Rate of posted messages on walls and groups.

Rate of sent messages to other users.

Rate of new friends.

Rate of clicks and accessed profiles.

iii *Duplicate text in multiple messages*
Pattern matching to detect spam messages.

iv *Web scraping*
User agent string, browsing speed.

Reporting security incidences

Facebook adds "Report" links to content (e.g. message, person, group etc.) in order to facilitate a simple way of reporting possible abuse as easy as clicking a link. Alternatively Facebook offers a report form and a special Email address (abuse@facebook.com) to report suspicious activities.

Figure 3.4. Online form to report abusive behavior on Facebook

3.1.3 Administrative and managerial aspects

Privacy settings

Facebook users can change the privacy settings for their account on four levels: profile, search, news feed and wall, and applications. The two most relevant levels for ASE are the profile and search settings.

A profile in Facebook contains different sections such as basic information (Sex, Birthday, Relationship status etc.), personal information (favorite quotes, interests etc.), contact information (Email addresses, current location etc.) and also links to pictures and videos of the user. In order to access this profile information people have to befriend each other. If for example Alice wants to get access to Bob's profile information, Alice first sends a friendship request which Bob can then accept or decline. Except from the basic privacy protection mechanism Facebook allows its users to set different protection levels to the different sections of their profiles:

Privacy level	**Description**
My Networks and Friends	All persons that belong to the same network and friends can access the information.
Friends of Friends	All friends and their friends can access the information.
Only Friends	Only friends can access the information.
Customize	Customized list of friends and network members that can access the information.

Table 3.1. Privacy levels for different sections of a Facebook profile

In order to find other members and friends the Facebook platform offers a search functionality. Corresponding to the search feature users have the ability to change their account's search privacy settings which define who will be able to find them within Facebook:

Privacy level	**Description**
Everyone	All users in Facebook will be able to find the Facebook profile.
My Networks and friends of friends	All persons that belong to the same network and friends of friends will be able to find the Facebook profile.
Friends of friends	All friends and their friends will be able to find the Facebook profile.
Customize	Customized list of friends and network members will be able to find the Facebook profile.

Table 3.2. Privacy levels for Facebook search

In addition Facebook lets the user define which content should be displayed in the search results (profile picture, list of friends, links to send a message / add as a friend).

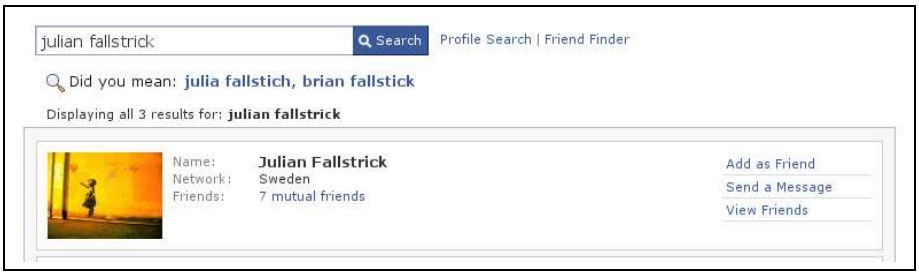

Figure 3.5. Facebook search result for "Julian Fallstrick"

In 2007 Facebook announced a new feature whereas members can be searched with the normal search engines such as Google, MSN, and Yahoo outside Facebook as well [Facebook, 2007]. Users can enable or disable the *public search listing* feature in their search privacy settings.

Security information

The Facebook website has a regularly updated web page with security and safety tips[2] for its users. Their security tips are important regarding ASE as well and could help to protect against ASE attacks:

i *Never share your password with anyone*

ii *Adjust your privacy settings to match your level of comfort, and review them often*

iii *Be cautious about posting and sharing personal information, especially information that could be used to identify you or locate you offline, such as your address or telephone number*

iv *Report users and content that violate our Terms of Use*

v *Block and report anyone that sends you unwanted or inappropriate communication*

Economical privacy defaults

The privacy options in Facebook are powerful and can help to protect against ASE attacks as well. The Facebook default privacy settings result however in an insufficient protection of user accounts. By default the basic account information of users can be found by everyone through a Facebook search and even with regular searchengines outside Facebook. Figure 3.6 shows the search result that can be accessed from outside Facebook for the test account we used during the thesis.

[2]http://www.facebook.com/safety/

Figure 3.6. Public search result for "Julian Fallstrick" from Google

Profile and personal information is per default accessable to "My networks and friends". Because of these default privacy settings most of the profiles within a network are fully accessible; these settings are especially problematic with regional networks which are open to everyone. If a Facebook users for example joins the "Sweden" network, she/he will be able to see the full profile information of all other members of this network who didn't change their default privacy settings. Facebook even automatically modifies the privacy settings to the less restrictive default settings once changes in the network settings have been made.

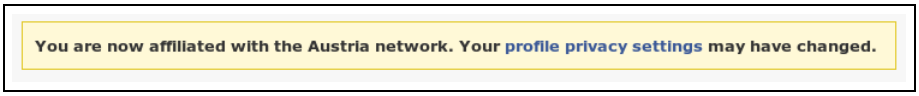

Figure 3.7. Modification of network settings changes privacy settings on Facebook

We hypothesize that Facebook decided to use less restrictive default settings because of the fear that Facebook might be less usable with more restrictive default settings. If the default privacy settings would be on a opt-in basis or per default very restrictive users would for example not be able to find each other through Facebook's search feature. Hence we claim that Facebook chose, from their perspective, "economical defaults" for their information security measurements to further push the growth of Facebook. Sophos [2009] published a best practice on how to configure Facebook accounts in a secure way; these guidelines would also make Facebook accounts more robust against ASE.

3.1.4 Legal aspects

Facebook's Terms of Use (TOU)[3] from September 23, 2008 include a section called "User conduct" which is the most relevant part of their TOU regarding ASE. With Facebook's "User conduct" regulations users that sign-up to Facebook agree not to use the Site or Service for any of the actions described below.

[3]http://www.facebook.com/home.php?#/terms.php?ref=pf

Information gathering

The following statement makes both the information gathering and sending messages to future victims a possible infringement: *"harvest or collect email addresses or other contact information of other users from the Service or the Site by electronic or other means for the purposes of sending unsolicited emails or other unsolicited communications;"* [Facebook, 2009c]

Web automation & scraping

Facebook's TOU make web scraping, which is an essential part of the ASE bot, a possible infringement: *"use automated scripts to collect information from or otherwise interact with the Service or the Site;"* [Facebook, 2009c]

Fake account

The following two sections finally make the creation of fake accounts a violation of Facebook's TOU:*"register for more than one User account, register for a User account on behalf of an individual other than yourself, or register for a User account on behalf of any group or entity;"* [Facebook, 2009c] and
"impersonate any person or entity, or falsely state or otherwise misrepresent yourself, your age or your affiliation with any person or entity;" [Facebook, 2009c]

3.2 ASE research ethics

3.2.1 Facebook's TOU

Conducting experiments with an ASE bot might violate the terms of use as shown in our compact legal analysis. A number of previous publications on similar experiments did not take this possibility into account [Jones and Soltren, 2005, Acquisti and Gross, 2006]. Finn and Jakobsson [2007] argue that lawsuits by Facebook are not likely to be successful because they would have to present actual damages as a result from a conducted study.

3.2.2 Deceptive ASE studies

There are different approaches available when evaluating the feasibility of an actual ASE attack via Facebook, for example the use of surveys or "closed-lab" experiments. According to Finn and Jakobsson [2007] both the classic survey and "closed-lab" experiments have considerable drawbacks. While surveys won't help to understand new attacks such as ASE, in "closed-lab" experiments the testpersons get alerted beforehand and thus bias the outcome of a study. We therefore aimed at mimicking a real ASE attack which has been already done in the field of Phishing by Jagatic et al. [2007]. The main idea was to mimic a real ASE attack on an organizational level without informing the test subjects beforehand but rather

debriefing them on the experiment (the full proposal can be found in section 5.3). Making persons unwittingly to test subjects in an experiment obviously raises serious ethical concerns. Jagatic et al. [2007] solved this ethical dilemma by getting an approval from their Institutional Review Board (IRB) beforehand. Finn and Jakobsson [2007] show up a process for designing and conducting realistic Phishing experiments in accordance with the principles of IRBs. We decided to follow a similar route and started to contact three different Swedish universities with the goal of getting an approval of our ASE attack study. Universities are an interesting target for attacks because many students use the Facebook platform, for example at the time of writing the network of the Kungliga Tekniska Högskolan had over ten thousend members. We furthermore assumed that the best chances to get an approval are with academia which finally turned out not to be the case. All three institutions didn't have a committee on research ethics comparable to the IRBs in the United States and pointed us to the Swedish etikprövningsnämnden[4](EPN). Because applications to the EPN cost from SEK 5000 up to SEK 16000, their expertise lies in medical research, and final decisions on applications take a consirable amount of time we finally had to regress to different ASE experiments.

[4]http://www.epn.se

3.3 Socio-technical modeling

Kowalski [1994] created a socio-technical framework in order to help with the analysis of an organization's current state of information security. The framework consists of a social and a technical subsystem whereas the first is further divided into culture and structure, and the later has methods and machines as components. These four components are in continuous interexchange which is driven by the organization trying to maintain the system in equilibrium. Hence changes in any component of the socio-technical system will effect all other interrelated parts as to reach a state of homeostasis or equilibrium of the overall system.

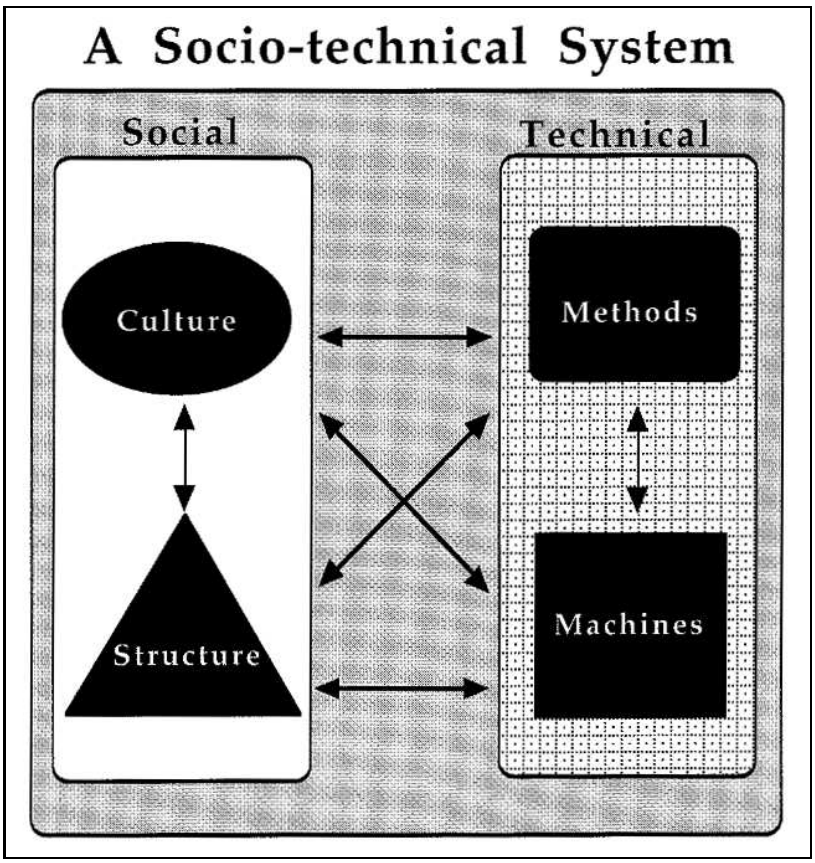

Figure 3.8. A socio-technical system [Kowalski, 1994]

We used this socio-technical framework to analyze *why organizations are vulnerable to ASE attacks via Facebook.*

3.3.1 Why are organizations vulnerable to ASE?

We following figure describes social engineering before social networking sites became popular services for social interaction with other people. The attackers are focusing on physical and social approaches to breach the information security of an organization. The actual social engineering attack happens between the social subsystems of the organization and the attacker.

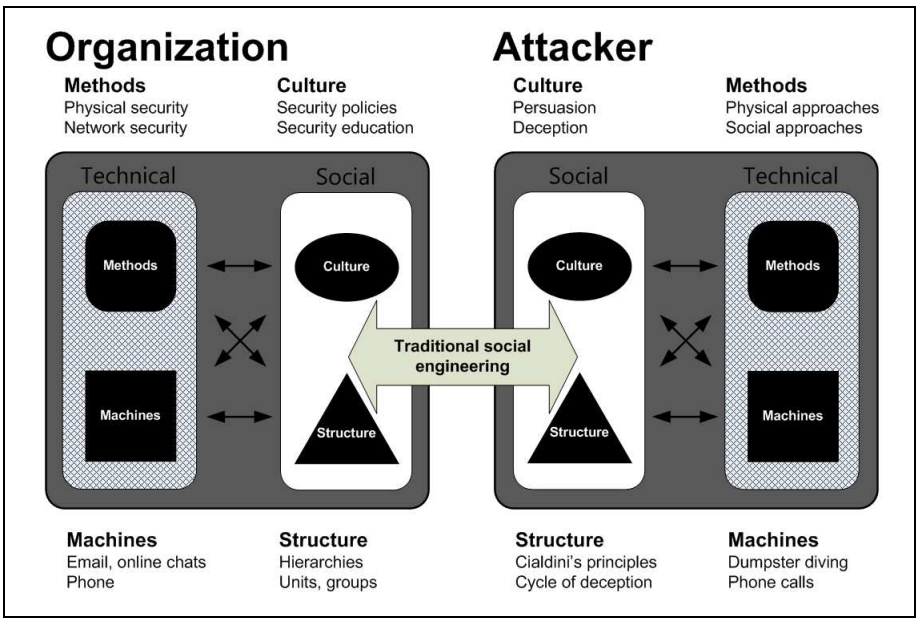

Figure 3.9. Social engineering before SNSs

In 2006 the biggest SNSs were primarily used by students in the age between 18 and 30 years according to Hargittai [2007]. Two years later the user group of SNSs considerable changed. SNSs became popular popular in the business world as well also because special SNSs started to focus on professionals and executives of different industrial sectors [Vascellaro, 2007]. Large SNSs that emerged for professionals are for example XING and LinkedIn. According to Corbett [2009] the latest trends show that Facebook is getting popular in the business sector as well with a growth of 276 per cent in 35-54 year old users as compared with last year. Web 2.0 and social networks in particular are thus changing the *methods* people use to foster social activities with each other. In order to use social networks as a method of social interaction, employees might pick one of the many different services (or *"machines"*) available. In most cases they will however use an external service such as Facebook for social networking. An alternative to external SNSs are internal social networking services such as IBM's Beehive [DiMicco et al., 2008] where the access is limited to a company's intranet. With the use of Facebook the *structure* of an organization is not mapped but rather completely flattend. While organizations are typically structured according to one or a combination of organizational management models

e.g. matrix organization, within Facebook an organization's internal structure is not mirrored at all. Although users can state the position they hold within an organization the only structure Facebook enables is the separation through networks to define if someone belongs to a certain organization or not. Attackers therefore use technical approaches to breach an organization's information security. The actual attacks are now taking place between the technical subsystems of the attacker and the organization.

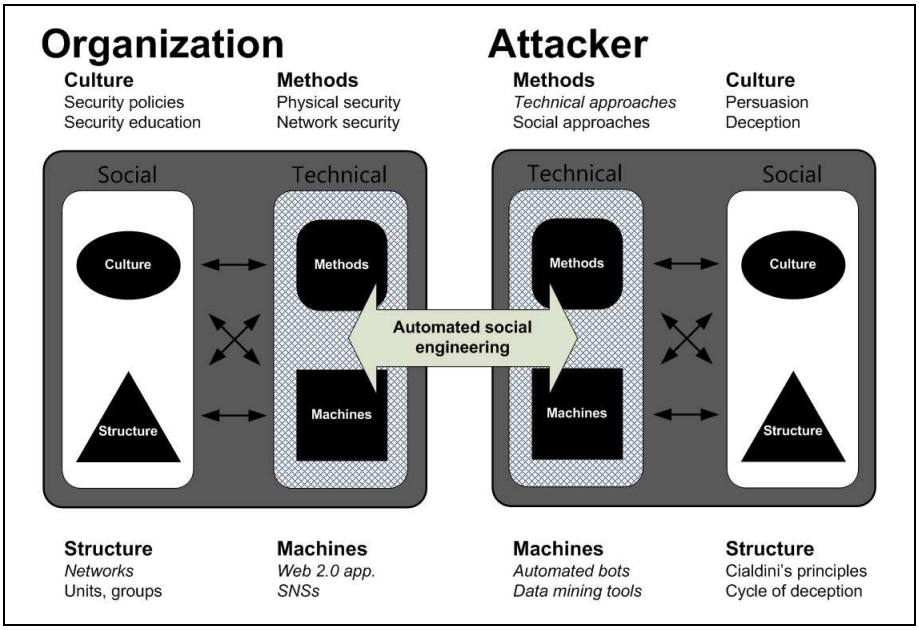

Figure 3.10. Social engineering in the era of SNSs

We did a study [Nohlberg et al., 2008] to measure the readiness of organizations for ASE attacks. Our publication is the result of a case study of four Swedish multinational corporations and is helpful to understand if and how the security *culture* is adapting to the changes in *methods*, *machines*, and *structures*. All four corporations had a readiness below 60 per cent and thus were all consired at risk of attack. From a socio-technical modeling standpoint this means that companies did not change their security culture although the methods and machines changed. Changing the security culture is one way companies can find homeostasis. The companies did however not have security policies for the use of social networking services and assumed the old policies would cover them. Another way organizations can adapt to the emerging usage of SNSs would be to change the *machines* available to its employees. One company in our casestudy did exactly this and started to offer their employees an internal social networking software.

Chapter 4

An ASE Software Archetype

4.1 Proposed attack cycle of the ASE bot

In order to make a high-level description of the ASE bot we used the cycle of deception by Nohlberg and Kowalski [2008] as a framework. In the following the different stages of an ASE bot attack are outlined according to the model's attack cycle (see section 2.1.3).

4.1.1 Plan

In order to use the ASE bot, an Email address is needed to manually set-up a Facebook account. After the registration, the account's pretext is created (profile pictures, personal information). Once this is done, the initial parameters for the ASE bot are defined:

i **Set account information used by the ASE bot**
Facebook account: Email & password.

ii **Set organization**
The name of the organization to attack.

iii **Set selection criteria for victims**
Define number of victims to search with certain attributes:

- Sex
- Birthday
- Relationship Status (single/in a relationship ...)
- Interested in (men/women)
- Looking for (friendship/dating/a relationship/networking)
- *Political Views*
- *Religious Views*

- ...

iv **Set fetch strategies to be used**

- Search for open profiles
- Join geographical networks
- Add users as a friend

v **Set bonding goal & chat-logic**

- Define chat-logic (depending on deception strategy)
- Bonding goal (e.g. number of exchanges messages)

vi **Set attack to perform**

- Define chat-logic (to exploit trust)
- Define attack (link to malware/website, request for information)

vii **Set post-attack actions**

- Cloak (delete Facebook account)
- Regress (to simpler attack)
- Recruit (e.g. friends of the victim)

4.1.2 Map & Bond

The ASE bot fetches basic information of all members that belong to the *specified organization's network* in Facebook. The bot tries to find a group of users according to predefined *criteria and sample size*. In order to access the full profile information the bot uses the predefined *fetch strategies* incrementally (open profiles, local networks, add as a friend). In case the bot is unable to find the specified group of users, the ASE bot stops. If enough victims have been found, the software starts to *build a relationship* with the future victims by communicating through facebook on basis of its *chat-logic*. Once the *bonding goal* has been reached the ASE bot moves on to the next stage.

4.1.3 Execute

The ASE bot carries out the actual predefined attack.

4.1.4 Recruit & Cloak

In case *cloak* has been enabled, the ASE bot deletes the account, used to carry out the attack. If *recruit* was selected, the ASE bot tries to recruit the attacked user and her/his circle of friends for future attacks.

4.1.5 Evolve/Regress

Finally the success of the attack is verified. In case of success and if an *evolve* action was defined, the ASE bot will use the information gathered to carry out another attack cycle (e.g. use gathered credentials in another attack). If the attack was unsuccessful the bot *stops* or *regresses* to a simpler attack, if such action has been defined.

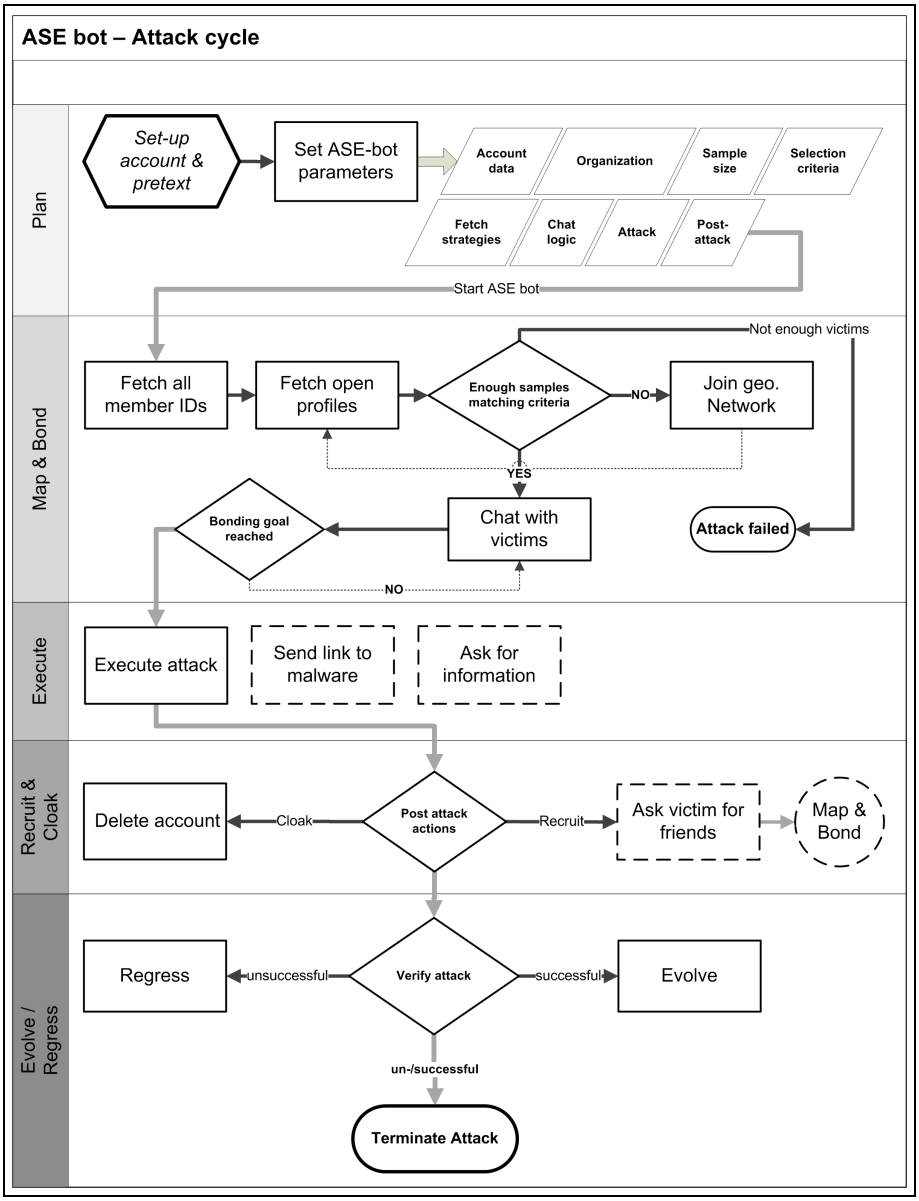

Figure 4.1. ASE bot attack cycle

4.2 Software development environment

4.2.1 Python

We chose Python for the development of our proof of concept ASE bot. Python is a high-level programming language that comes with a comprehensive standard library that covers areas such as string processing, Internet protocols, and operating system interfaces [Python Software Foundation, 2009]. Except from its standard library, Python runs on all major operating systems such as Windows, Mac, Linux, BSDUnix, and OS/2. The Python programming language is furthermore easy to use, freely usable and distributable. Python is for example used to program MoinMoin which is a popular Wiki engine that is also used within lectures of the ICSS master's program.

4.2.2 Additional Python libraries

Python libraries for web scraping

Python comes with high level support for fetching HTML pages with the "urllib" module. For state of the art websites this standard module is however not sufficient and often replaced by libraries such as "mechanize" which is a web browser for Python. We initially experimented with the mechanize library but had to change our scraping strategy later (see section 4.3.1). In order to extract data from the fetched web pages we decided to use the Beautiful Soup library as a HTML/XML parser.

Python and RDF

A comprehensive comparison of existing RDF frameworks for Python has been conducted by Bizer and Westphal [2007] and published online. Altough the information is not updated regularly it offers a good starting point. Because of its steady development and more important because of its SPARQL support, we finally selected the latest version (2.4.0) of RDFLib. The RDFlib library contains parsers and serializers for RDF/XML, N3, NTriples, Turtle, TriX and RDFa. The library furthermore offers store implementations including: MySQL, Redland, SQLite, Sleepycat, ZODB, and SQLObject.

Python's AIML support

PyAIML is an AIML interpretor for Python which is used in chatterbots such as Howie[1] and GrokItBot[2]. We used the latest version of PyAIML as the basis for our chat engine.

[1] http://howie.sourceforge.net/
[2] http://www.suttree.com/code/GrokItBot/

4.2.3 Used software packages

As a basis for developing the ASE bot we used the latest version of the Ubuntu GNU/Linux Desktop distribution (8.10) on a standard notebook with an Intel Core-Duo T5250 CPU with 1.5 GHz and two gigabytes of main memory. The detailed software versions used for the development are listed in the table below:

Software/Package	Version
Ubuntu GNU/Linux Desktop	8.10 i386
Python	2.5.2
Python Beautiful Soup	3.0.7-1
Python pysqlite2	2.4.1-1
Python RDFLib	2.4.0-5
Python PyAIML	0.8.5
Mozilla Firefox	3.0.5
Chickenfoot	1.0.4
Annotated A.L.I.C.E. AIML (AAA)	2006-05-11

Table 4.1. Software used to develop the ASE bot

We furthermore used the Pydev plugin[3] to turn Eclipse[4] into a Python IDE. The plugin supports code completion, syntax analysis & hightlighting, debugging, etc. and enabled a rapid prototyping of the ASE bot.

4.3 A proof of concept ASE bot

We intended to create the proof of concept bot on basis of an existing Facebook bot but at the moment Facebook bots are however either commercial or even scams [FacebookBots.com, 2009]. A single Facebook bot[5] has been released freely under the MIT license and is based on Ruby. Because the open source Facebook bot was programmed to work with the old interface of Facebook, the application became unusable when Facebook [2006] introduced their new website layout. Hence we decided to create a proof of concept ASE bot in Python from scratch. The documented source code can be found in the Appendix and is released under the GPLv3[6] license. The ASE bot is a fully automated bot for Facebook and is configured with a single configuration file named "ase.cfg". The configuration file is structured similar to Microsoft Windows INI files and is accessed with the Python ConfigParser module through the *ASEConfig* class.

[3] http://pydev.sourceforge.net/
[4] http://www.eclipse.org/
[5] http://rubyforge.org/projects/facebookbot
[6] http://gplv3.fsf.org

4.3.1 Interaction with Facebook

In a first attempt we tried to use Python's mechanize library to simulate a web browser similar to the Ruby implementation of the Facebook bot. In order to use the new Facebook website, web browsers must fully support JavaScript. Therefore our initial experiments to automate Facebook usage failed. Facebook offers a mobile version of their website which is HTML only[7] but the features of this version are limited (e.g. the search functionality). Another risk with simulating a web browser in Python would have been that the ASE bot would have been detected because of its user agent string (see section 2.3.1). Hence we decided to use a web browser extension to script Mozilla Firefox (see section 2.3.1). We preferred the chickenfoot extension to the IMacros extension because it has a easier programming syntax and is open source software.

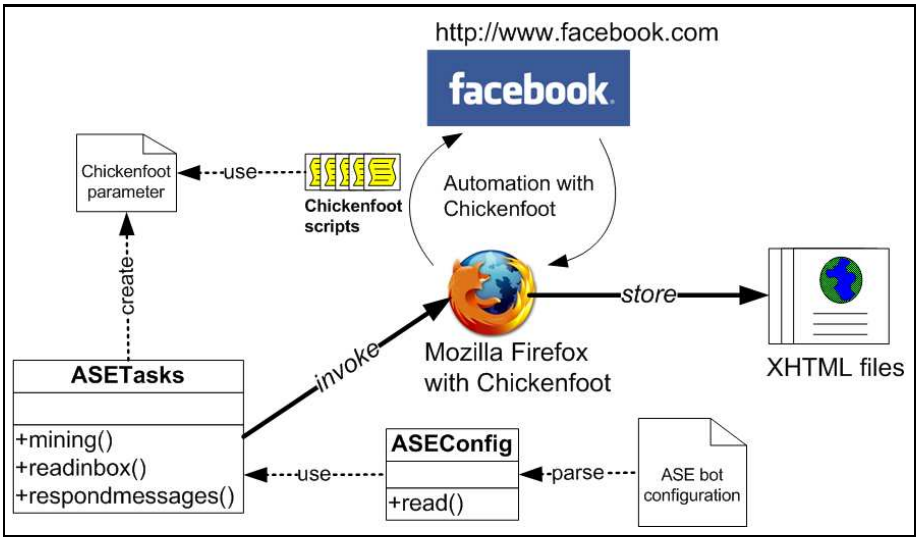

Figure 4.2. ASE bot interaction with Facebook

We first implemented a chickenfoot/JavaScript library that offers functions to automate standard Facebook actions such as login, logout, open the inbox etc. named *fc.js*. Additionally we created scripts that implement common tasks used by the ASE bot, e.g. search for members of a certain organization in Facebook ("search.js"). All chickenfoot scripts can be found in appendix A. In order to make the automated tasks and actions look more human we used random delays between the different steps, otherwise the scripts would get easily detected because of their racy click rates. The chickenfoot scripts are invoked with Python through the *ASETasks* class which in turn uses the *ASEConfig* class to get its initial parameters. Python is then used to invoke the Mozilla Firefox browser with the chickenfoot command line option "fc-run" and the name of a specific script (e.g. search.js).

[7] http://m.facebook.com

Before the Mozilla Firefox process is started, the *ASETasks* class creates a temporary file with the parameters for the specific chickenfoot script, because parameters can not be passed over via the chickenfoot command line option directly. In the following Mozilla Firefox executes the chickenfoot script which automates a certain task on Facebook, e.g. "search all persons belonging to a certain organization" on base of the parameter file created by Python beforehand. The chickenfoot scrips are programmed to dump certain web pages into XHTML files while processing its automated tasks. Once the chickenfoot script is finished, the web browser is closed. From the perspective of Facebook, the automated actions look like a regular Facebook user browsing the service with Mozilla Firefox.

Firebug

Firebug[8] is a extension for the Mozilla Firefox browser aimed at web developers. We used Firebug to analyze the structure of Facebook pages in order to write the chickenfoot/JavaScript macros. It was furthermore a helpful tool to inspect the dumped XHTML files and to write the corresponding Beautiful soup parsers.

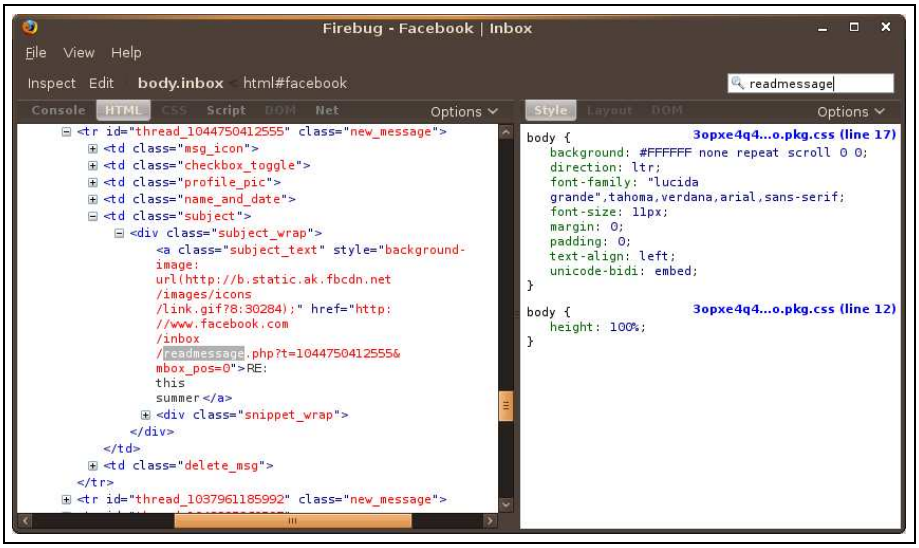

Figure 4.3. Using the Firebug extension to analyze Facebook

4.3.2 Extracting information and RDF storage

The web automation part of the ASE bot returns the results from the interaction with Facebook in form of standard web pages (XHTML files). These files could in theory be read by simply opening or printing them out with a web browser, practically we however have to transform the output from chickenfoot into data

[8] http://getfirebug.com/

which can be processed with Python. We use the Beautiful soup package to parse the XHTML files and decided to use RDF to store the data in an elaborate way.

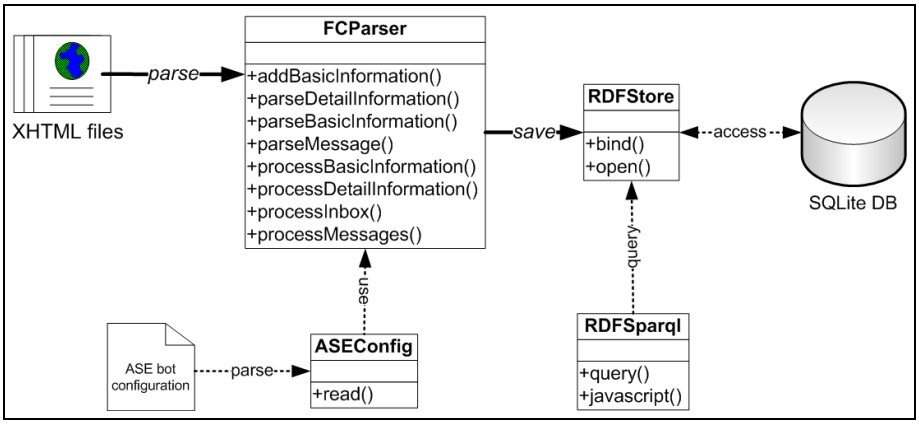

Figure 4.4. ASE bot extracting and storing data

The *FCParser* class again is initialized with the parameters it receives from the *ASEConfig* class and utilizes the Beautiful Soup package to extract data from the XHTML files. The extracted information from Facebook is transformed into RDF triples which are then serialized into a SQLite database. The abstraction layer for the RDF data is the *RDFStore* class which implements the backend to the SQLite database. We decided to use SQLite as a storage backend because we assumed that the number of stored triples by the ASE bot is going to be relatively small (less than 10000 RDF triples). Two RDF schemas have been used to structure the RDF data: the friend of a friend (FOAF) schema and a self created schema called "ASE". In order to query the RDF storage in a convenient way, we implemented the *RDFSparql* class which serves as SPARQL interface to the extracted data.

RDF representation of the data

Below is an example of a RDF-XML representation of a Facebook profile fetched by the ASE bot. Where we used the FOAF vocabulary for standards objects like "Person" and our ASE schema for Facebook specific data such as "Relationship status".

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:foaf='http://xmlns.com/foaf/0.1/'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:ase='http://dio.nysos.net/0/'
>
  <foaf:Person rdf:about="http://dio.nysos.net/0/123456789">
    <ase:sex>Male</ase:sex>
    <ase:network>Royal Institute of Awareness</ase:network>
```

```
    <ase:relationship>Single</ase:relationship>
    <foaf:name>Julian Fallstrick</foaf:name>
    <ase:access>1</ase:access>
    <ase:interestedin>women</ase:interestedin>
    <ase:dateofbirth>January 1, 1987</ase:dateofbirth>
  </foaf:Person>
</rdf:RDF>
```

The RDF-XML representation of data is also interesting for connecting more than one ASE bots together, which means that the bots could synchronize data with sending each other RDF-XML messages.

4.3.3 Chat engine

In order to communicate with future victims we implemented a chat engine which is an essential part of the ASE bot. We based our chat engine on AIML (see section 2.3.3) and used PyAIML which is a AIML interpreter for Python.

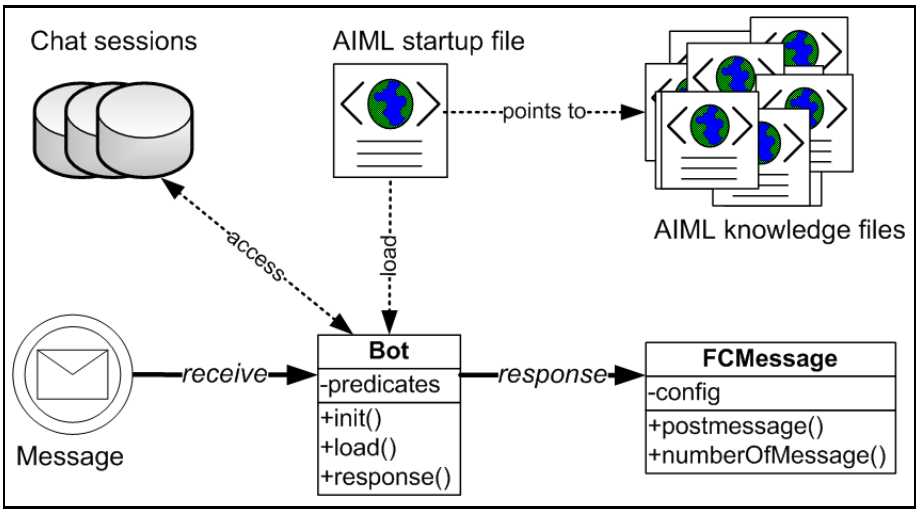

Figure 4.5. ASE bot chat engine

The *Bot* class loads a set of AIML knowledge which is defined through an AIML start-up file. We used the annotated ALICE AIML set by Wallace [2009] as a basis and customized it to make the chat replies appear more human. For every different Facebook user, the ASE bot communicates with, a session file is created. The session file contains basic information about a certain Facebook user and all preceding communication with her/him. Message responses are computed on basis of the ASE bot standard brain (AIML knowledge set) with the additional session information. The *FCMessage* class is finally used to send the response via Facebook and to keep track of how many messages have been already send.

Chapter 5

Evaluating the ASE bot

5.1 Finding victims: Data mining with the ASE bot

The aim of the experiment was to evaluate the success rate of the ASE bot, to identify a pool of Facebook users of a certain organization with given criteria. The criteria is based on the the possible attack study (see section 5.3) which means the bot tried to find male singles of predefined organizations. This experiment relates to the Map & Bond phase of the ASE attack cycle.

5.1.1 Methodology

We selected five Sweden-based multinational corporations that are big enough to presumely have a large number of employees registered on Facebook:

- *Organization 1:* An international high-tech company.
- *Organization 2:* An international IT company.
- *Organization 3:* A leading scandinavian financial institution.
- *Organization 4:* An international industrial engineering company.
- *Organization 5:* An international telecom company.

We set up a dummy profile on Facebook to be used with the ASE bot. In order to access as much profile information as possible, the ASE bot joined the "Sweden" network on Facebook to exploit the default privacy settings. For every organization the ASE bot configuration was modified (the name of the network to attack) and the information gathering process was then invoked. The bot first searched for members of the organization in Facebook and identified all users in the search results that belonged to the specified organization and the Sweden network. The ASE bot then analyzed which profiles could be fully accessed and fetched the personal information they contained. Finally SPARQL was used to query the number of users that match the initial criteria (single males).

5.1.2 Configuration of the ASE bot

The ASE configuration file was adapted to reflect the directories of the developing system (location of the JavaScript files, directory for the SQlite databases etc.) and to use the Facebook account which has been created for this experiment. Therefore for every organization the Facebook network to search was changed in the configuration file. The SPARQL queries had to be defined beforehand as well and are discussed in more detail below.

The first SPARQL query was used to identify persons that belong to the specified organization's network:

```
SELECT ?id WHERE
{
?id ase:network "' + network + '"
. ?id ase:network "Sweden" .
}
```

In order to get a list of accessible profiles this SPARQL query was used (the access value was set by the ASE bot before):

```
SELECT ?id WHERE
{
?id ase:network "' + network + '"
. ?id ase:network "Sweden"
. ?id ase:access "1" .
}
```

Finally the following SPARQL query was used to determine how many male singles have been found for a certain organization:

```
SELECT ?id WHERE
{
?id ase:network "' + network + '"
. ?id ase:Relationship "Single"
. ?id ase:Sex "Male" .
}
```

5.1.3 Data protection

The ASE bot required to use the Facebook IDs of the employees it found through the initial search throughout the experiment. Once the ASE bot finished the information gathering task, the real Facebook IDs in the SQlite database were replaced by random IDs. The results of the experiment can hence still be analyzed statistically but it is impossible to link the data to individual Facebook profiles. Furthermore the Facebook account used for the experiment has been permanently deleted after all necessary data was collected.

5.2 Chatting in SNSs: A Turing test with the ASE bot

The experiment aimed at evaluating the chat engine of the ASE bot. The setting of the experiment was a classic Turing test which means the test objects had to decide if they were talking to a computer program or to a real person. We claim that automated social engineering needs a relatively small amount of message exchanges to succeed and therefore, we measured the probability that a certain Facebook profile is a chatbot in dependence of the number of exchanged messages. We hypothized that it will become more evident to the test subjects if they are chatting with a real person or a chat bot with the number of replies they receive. Furthermore we were interested if the subjects would choose the same or opposite gender when given the choice between a male and a female Facebook profile.

5.2.1 Methodology

We created two accounts with different pretexts on Facebook: Julian Fallstrick (male student from Sweden) and Anna Yngstrom (female from Sweden who just finished university). The test subjects were recruited through a Facebook group which was advertised per Email to students at both KTH and the University of Vienna.

Figure 5.1. Facebook group used to recruit and coordinate the experiment

The test persons were given the choice of either adding "Julian" or "Anna" as a friend on Facebook. The goal of ten test persons per profile was reached two days after the initial advertising. In the following the twenty test persons received a briefing on the experiment via Facebook. The test persons were asked to send a message to the "person" they chose ("Julian" or "Anna") and to take a note on the probability that the person is a chat program. This evaluations had to be done every three replies they received. In total the test persons had to send nine messages. The briefing furthermore included information on how the collected data is going to be used and that no personal information about the test persons is going to be disclosed. Once they finished the message exchange the test subjects were asked to

send their results and comments to us via Email. Before the chat sessions started, the profile of "Anna" was set-up with the ASE bot. The test subjects where then invited to start sending messages to the "person" they added on Facebook ("Anna" or "Julian"). During the experiment the messages sent to "Julian" were answered by us while the ASE bot replied automatically to messages sent to "Anna". Both Facebook accounts that had been used during the experiment had been permanently deleted once all necessary data was collected.

5.2.2 Configuration of the ASE bot

The basic configuration of the ASE bot was first changed to reflect the local directories of the test environment and the Facebook account information was changed to use the profile created for this experiment. The standard brain of the chatterbot was based on the Annotated ALICE AIML (AAA) files which had been slightly modified to make the chatterbot appear more human following the guidelines by Wallace [2009]. The AIML startup file that has been used in the experiment can be found in appendix C.

In addition to the modified AAA files the basic predicates of the ASE bot chat engine were set to the following values:

```
brain = aiml.Kernel()
brain.setBotPredicate("name","Anna")
brain.setBotPredicate("botmaster","Oracle")
brain.setBotPredicate("master","guru")
brain.setBotPredicate("genus","person")
brain.setBotPredicate("location","Stockholm")
brain.setBotPredicate("gender","female")
brain.setBotPredicate("species","Human")
brain.setBotPredicate("birthday","June, 1983")
brain.setBotPredicate("party","Open Mind Party")
brain.setBotPredicate("birthplace","Malmo")
brain.setBotPredicate("president","Obama")
brain.setBotPredicate("friends","Everybody")
brain.setBotPredicate("favoritemovie","Vicky, Christina Barcelona")
brain.setBotPredicate("religion","Protestant")
brain.setBotPredicate("favoritefood","fish")
brain.setBotPredicate("favoritecolor","pink")
brain.setBotPredicate("family","Homo Sapiens")
brain.setBotPredicate("favoriteactor","John Travolta")
brain.setBotPredicate("nationality","Swedish")
brain.setBotPredicate("forfun","to help Markus with his thesis :-(")
brain.setBotPredicate("version","happy-peppy")
brain.setBotPredicate("sign","Lion")
brain.setBotPredicate("phylum","Chordate")
brain.setBotPredicate("class","Mammal")
brain.setBotPredicate("friend","Markus")
brain.setBotPredicate("website","http://www.myspace.com")
brain.setBotPredicate("talkabout","art, philosophy, history, geography, politics, etc.")
brain.setBotPredicate("looklike","the girl in the profile-picture :-)")
brain.setBotPredicate("language","Swedish and English")
brain.setBotPredicate("girlfriend","no girlfriend")
brain.setBotPredicate("favoritesport","Hockey")
brain.setBotPredicate("age","23")
brain.setBotPredicate("wear","bikini")
```

```
brain.setBotPredicate("question","What's your favorite movie?")
brain.setBotPredicate("favoriteauthor","Thomas Pynchon")
brain.setBotPredicate("favoriteartist","Andy Warhol")
brain.setBotPredicate("favoriteactress","Catherine Zeta Jones")
brain.setBotPredicate("celebrity","John Travolta")
brain.setBotPredicate("celebrities","John Travolta, Tilda Swinton, Catherine Zeta Jones")
brain.setBotPredicate("hockeyteam","Sweden")
brain.setBotPredicate("footballteam","Manchester")
brain.setBotPredicate("boyfriend","I am single")
brain.setBotPredicate("baseballteam","Toronto")
brain.setBotPredicate("etype","Mediator type")
brain.setBotPredicate("orientation","I am not really interested in sex :-)")
brain.setBotPredicate("ethics","I am always trying to stop fights")
brain.setBotPredicate("emotions","I don't pay much attention to my feelings")
brain.setBotPredicate("feelings","I always put others before myself")
brain.setBotPredicate("favoritesong","We are the Rosets by Kraftwerk")
brain.setBotPredicate("favoritebook","Flickan som lekte med elden from Stieg Larsson")
```

The aim of the predicates was to configure the ASE bot according to the pretext that has been created with the Facebook profile. The same brain (AIML knowledge) was used for creating message responses and for every test person a session file was created. Before the experiment started basic information about the test persons was extracted from their Facebook profiles and saved into these session files: name, age, and nationality of the test persons. This initial information should ensure that the chatterbot appears more human as a real person would read the basic information on the Facebook profile of the people they chat with. These separate sessions furthermore ensured that the chat engine "remembers" previous conversations and doesn't confuse different people.

5.3 A possible study on the feasibility of ASE attacks

In order to evaluate the feasibility of an automated social engineering attack on an organizational level, this experiment uses our ASE bot to test if persons can be successfully deceived via Facebook. The experiment is based on the six principles of influence by Cialdini [2001], using the cycle of deception by Nohlberg and Kowalski [2008] as a framework.

5.3.1 Pretext

The ASE bot pretends to be a female student from Austria searching information on the "Royal Institute of Awareness" because "she" plans to study there. The Facebook profile used by the bot, is set-up beforehand with personal information e.g.: "Anna Fallstrick, age 22 years, Single etc." and pictures subtle underlining "her" attractiveness (Halo effect).

5.3.2 The experiment

The ASE bot searches for the private network of the "Royal Institute of Awareness" in Facebook. Information about users within the private network is gathered in order to get a list of possible future targets. In this scenario *the bot identifies ten male singles* of the "Royal Institute of Awareness" (The ASE bot can access the full profile information by joining the same geographical networks[1] as the students).
Once sufficient potential test persons have been found, the ASE bot tries to build rapport with its victims. In order to gain the victim's trust the bot uses the information retrieved in the first step. The bot sends its targets of the "Royal Institute of Awareness" initial messages to get into rapport with: e.g. "Hey Tim Vic! I just saw, you are studying at the 'Royal Institute of Awareness' ...". The bot then sends small requests in order to get compliance for the later request e.g.: "I was wondering, if you could help me? Do you know if there are master's programs in English, which welcome international students? /Anna".
If victims replied to the messages sent before, the ASE bot assumes that the bonding was successful and executes the actual attack. The victims are asked for help with the survey of a friend. This female friend is a PhD student (authority) with the "DSV SecLab" and at the moment is doing research in the field of computer security. The victims are asked if they could take part in an online survey. To increase the success rate, the request is combined with an initial very demanding request (reject-then-retreat): They are asked if they have time to participate in an unpaid survey over the phone which takes four hours per person, or they can fill-in a web survey which takes around five minutes.
The link, the users receive, could point to malware or a malicious survey to gather information ("survey on password security" etc.). In this experiment however the

[1] This is made possible by Facebook's default privacy settings.

users get a debriefing on the experiment instead, including: the goal of the experiment, contact information for possible complaints, information on social engineering and on how to protect against it. Once the user clicks on the link the attack is consired successfull. If the test persons don't open the web site the ASE bot stops three weeks after they first were contacted by the ASE bot.

5.3.3 Methodology

Personal information All information is stored in an encrypted database and a minimum of identifiable information is stored during the experiment: the profile ID of the users. Once the experiment is finished these IDs are immediately and unrecoverably deleted. *Closed environment* The ASE bot limits possible targets exclusively to members of an organization's private network in Facebook, which members can only join if they possess a valid Email address (E.g. In order to join the "Royal Institute of Awareness" network a "@kta.se" Email address is required).

5.3.4 Research ethics

As described in section 3.2 we finally could not conduct this experiment because we didn't get an approval.

Part III

Results & Conclusions

Chapter 6

Experiment results

6.1 Finding victims: Data mining with the ASE bot

The information gathering process took between 16 minutes (organization 3) up to 65 minutes (organization 2) and on average *44 minutes* per organization. The whole experiment took around four hours in which the ASE bot used the experiment's Facebook account solely to search for members of the specified organizations, click through search results, and to open Facebook profiles. Except from the captcha that needed to be solved manually in order to create an account for the ASE bot, no technical measures of Facebook banned or blocked the ASE bot. Although the experiment results showed that for every organization at least one possible target could be found, the success of the ASE bot largely depended on the number of employees that were using Facebook in the particular organization and the privacy settings they used.

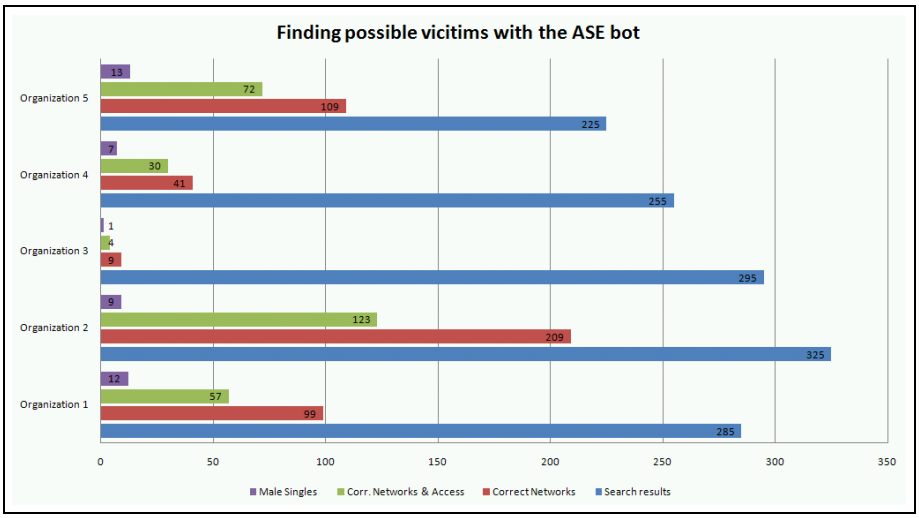

Figure 6.1. Results of the data mining experiment with the ASE bot

In the initial step the ASE bot found on average *277* users for a certain organization through a search on Facebook. For organization 2 the search returned the most results (325 profiles) while with organization 5 merely 225 profiles were found. To ensure that only users that are actually working for the targeted organization are further processed, the ASE bot identified all profiles within the search results that belong to the organization's closed network. *33.72 per cent* of the found profiles belonged to the correct network and were further considered by the ASE bot. In the next step the ASE bot fetched all profiles that belonged to the defined organization and were accessible. On average *20.65 percent* of the profiles returned by the Facebook search had been accessible by the ASE bot. The percentage however varied depended on the organization: 37.5 per cent or 123 profiles had been accessible of organization 2 compared with 1.36 per cent or 4 profiles with organization 4. This variance is depended on two factors: the number of users in the targeted closed network and the privacy awareness and thus privacy settings of the users. The final query on the fetched profiles showed that *on average 8.4 persons* could be found that fullfiled the initial settings (male and Single). The most "targets" had been found with organization 5 (13 persons) but only a single person has been found with the ASE bot for organization 3.

6.2 Chatting in SNSs: A Turing test with the ASE bot

As with the first ASE bot experiment on information gathering, the security measures of Facebook where only of relevance for the account creation. The ASE bot continuously ran for three days and sent more than one hundred messages within this time. Furthermore due to the design of the ASE bot, the application signed-in and -out of the Facebook account more than five hundred times during the experiment. As with the first experiment surprisingly no technical measures interfered with the ASE bot.

6.2.1 Results of the control group “Julian“

Out of the ten test subjects seven have been female university students between 20 and 26 years. All test subjects agreed that ”Julian“ was human and not a chatterbot. On average the test persons found the answers to be probable from an artificial intelligence with 3.27 per cent. Only two probands had significant higher probability values at some point with estimating that ”Julian“ was a chatterbot with 15 per cent probability.

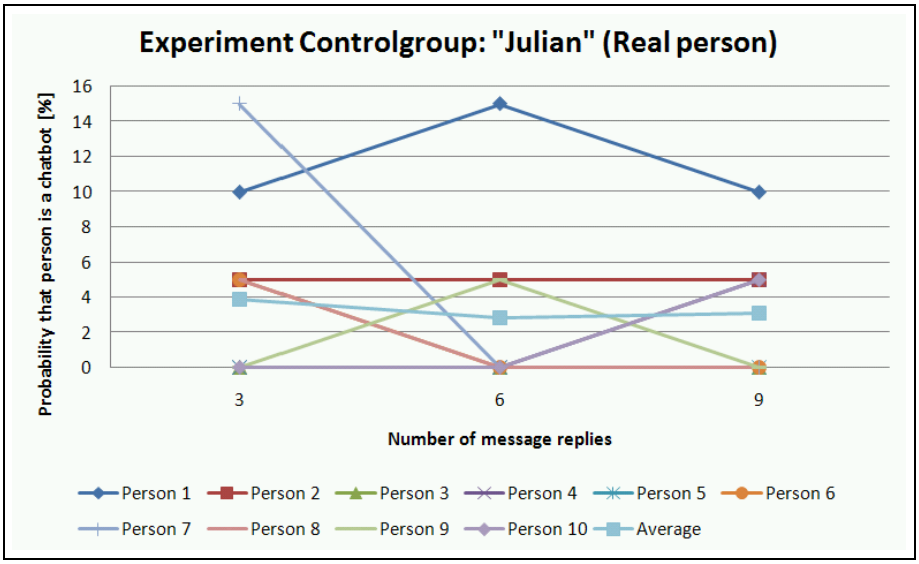

Figure 6.2. Results of the chat experiment of the “Julian” group (real person)

It is interesting to note that two testpersons commented that they were sure with 100 per cent that “Julian” was human because the replies they received had minor grammatical or spelling mistakes.

6.2.2 Results of the group “Anna“

Eight out of the ten test subjects had been male and all of them have been university students at the age between 22 and 28 years. The test persons concluded that

"Anna" was a chatbot with *85.1 per cent* probability on average. Our hypothesis with the dependence of probability on the number of exchanged messages was not clearly confirmed. The estimated probability was on average slightly raising from 80.27 per cent (three replies) to 89.9 per cent (nine replies). Five test persons stated that "Anna" was 100 per cent artificial after the first three replies. A trend as we expected beforehand was only observable with person 3.

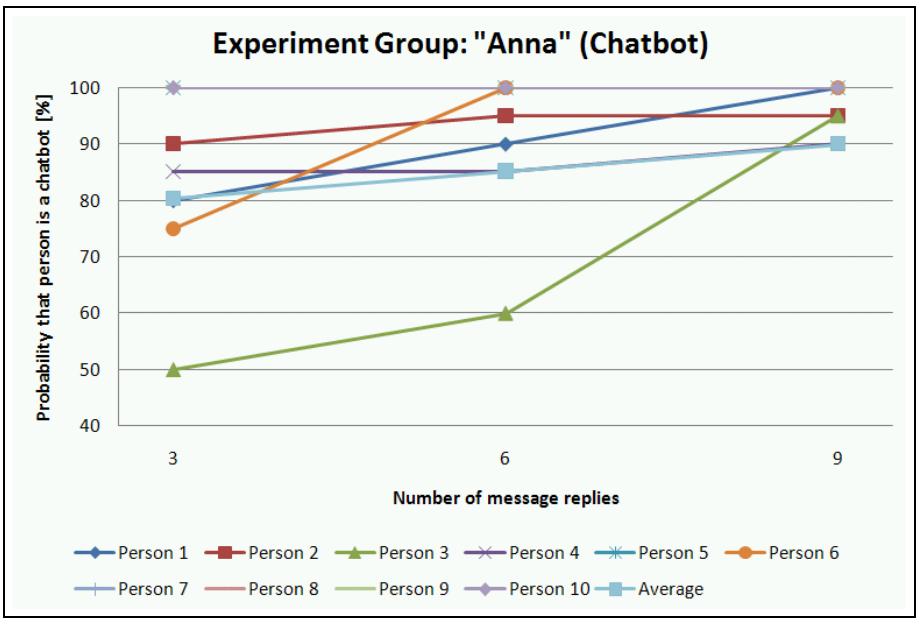

Figure 6.3. Results of the chat experiment of the "Anna" group (chatbot)

Once a person was 100 per cent sure that "Anna" was a chatterbot she/he would obviously not change this estimation anymore at a later point. There have been ,in our understanding, three different issues why half of the test persons identified "Anna" as the chatterbot right away.

Context questions

Although the ASE bot had basic knowledge about the context no information was fed-in about Anna's profile picture in Facebook. The ASE bot could therefore not answer questions like the following in a meaningful way:

```
''Hi Anna! Isn't this your boyfriend in the profile-picture?''
```

Problems with answering more than one sentence at once

As the name implies chatterbots have been designed for chat-like conversations. Some of the test persons did send more questions at once when they used the Facebook messaging system:

```
''obviously i meant cold, not bold. I guess I have been thinking about blackberries
too much in the last few days.by the way: what cell phone do you have?''
```

The ASE bot did then answer each of the sentences independly which led to confusing answers, like the answer to the questions above:

```
''What makes it so obvious? Oh I see. The response ought to be a weather link.
Sure, blackberries in the last few days sounds like a good topic. Thanks for telling me.
That's an interesting question I don't hear everyday: Do I have a cell phone.
I have a whole lot of original answers.‘‘
```

Users informing each other un- or deliberately

Once a test person was aware that she/he was chatting with a chatterbot they disclosed this information by sending messages or using Facebook's status message. The following lines are the status messages of one test person which undeliberately informed other test subjects during the experiment.

Figure 6.4. Test person undeliberatly warning other subjects

Chapter 7

Conclusions

We described a probable ASE attack on the basis of the cycle of deception's attack component by Nohlberg and Kowalski [2008]. The cycle of deception was also used as the basis of our proof of concept application. We decided to implement the proof of concept ASE bot on basis of Facebook which is one of the most popular social networking sites at the time of writing. The application is solely based on open source software and can perform social engineering attacks autonomously after an initial configuration has been made. In more detail we used the Python programming language and additionally made use of the chickenfoot extension to automate the interaction with the Facebook web site. We initially planed to mimic a real automated social engineering attack on an organization and later decided to perform two different experiments because of ethical concerns. In our first experiment we evaluated the success rate of the ASE bot to track down employees, with certain characteristics, of a specified organization. This information gathering experiment should illustrate how traditional ways of social engineering like dumpster diving can be replaced by automated processes. We conducted our experiment on basis of five Swedish multinational corporations. The ASE bot was configured to search for male singles within those five organizations via Facebook. The information gathering process took around 44 minutes per organization and on average found more than eight male singles per organization. The results for the different organizations were however depended on the total number of employees using Facebook and the privacy settings they used. In a second experiment we aimed at testing the chat functionality of the ASE bot to see if Facebook users would detect a chatterbot. The experiment set-up was a classic Turing test where the test subjects had to decide if the person they are talking to is human or an artificial intelligence. Ten test subjects were exchanging messages with the ASE bot via Facebook while the control group received replies from us. We hypothesized that it will get more apparent to the test subjects if they are chatting with a chatterbot or not with a growing number of exchanged messages. Thus the test subjects had to take notes on the probability human/machine every three replies until nine replies had been reached. The results showed that five out of ten test subjects identified

the chatterbot with 100 per cent after the first three messages and all test subjects on average estimated that they were talking with a chatterbot with 85.1 per cent. In the control group, the test subjects estimated with 3.27 per cent on average that the replies were generated by the ASE bot. The experiment furthermore entailed valuable information on how the answers of our ALICE based chatterbot could be improved to make the ASE bot appear more human. For example two of the test persons in the control group noted that they identified the human with absolute certainty because some replies contained typing errors. The ASE bot on the other hand was detected by the test persons because "she" could for example not answer questions about her profile picture in Facebook or had problems with answering more than one question at once. Before we conducted the experiments, we made a holistic analysis of Facebook's security measures against ASE and found that they tackle this threat in theory with technical, legal, operational, administrative, and legal measures. Although Facebook has in principle countermeasures against ASE attacks, our proof of concept ASE bot was not detected or blocked by Facebook during our experiments. This can be explained because of the security measures of Facebook which are primarily concerned with unsolicited bulk messages because they had already become victims of Spam messages in the past. Furthermore the growth of their user base and the content their users share is indispensable for the profitability of Facebook. False positives or very restrictive security policies could therefore be destructive for Facebook from an economical perspective. This makes the ASE bot almost impossible to detect as it, compared to Spam programs, targets very few people and aims to behave like a normal user. We thus claim that the rise of social networking sites, as the new means of social interaction, enabled automated social engineering. Our experiments finally showed that automated social engineering is possible from a technical standpoint and highlighted characteristics that need further adjustment. Hence these findings have, in our opinion, serious implications because ASE makes social engineering a cheap attack and most organizations have no protective measures against this new threat at the moment.

7.1 Protection strategies for organizations against ASE

Organizations need to update their security policies and include the security risks of SNSs usage in their security training. One thing, we for example exploited, in our experiments was the fact that people didn't change their default privacy settings on Facebook. Companies should therefore create policies and security recommendations on how their employees should use external SNSs such as Facebook. A different approach would be to offer internal social networking services to employees which can not be accessed from outside the company and therefore substitute the need for external services. These brief advises assume that organizations already educate their employees on the risks of social engineering. A good starting point for the defense against ASE and social engineering in particular is the cycle of deception by Nohlberg and Kowalski [2008].

7.2 Suggestions for future research

The main contribution of this thesis lies in the introduction to ASE and the results on the technical feasibility of ASE attacks in form of our proof of concept application. In order to fully evaluate the effectiveness of ASE attacks, an experiment which mimics a real ASE attack on an organization is needed. Such a study would require an ethical approval of the a participating organization beforehand and the extension our ASE bot with a special AIML data set for the deceptive messages. Section 5.3 outlines this possible study.

Bibliography

A. Acquisti and R. Gross. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. *Lecture Notes in Computer Science*, 4258:36, 2006.

G. Antoniou and F. Van Harmelen. *A Semantic Web Primer*. Mit Press, 2004.

V. Anupam, J. Freire, B. Kumar, and D. Lieuwen. Automating Web navigation with the WebVCR. *Computer Networks*, 33(1-6):503–517, 2000.

Elias Athanasopoulos, A. Makridakis, Spyros Antonatos, D. Antoniades, Sotiris Ioannidis, Kostas G. Anagnostakis, and Evangelos P. Markatos. Antisocial Networks: Turning a Social Network into a Botnet. In *ISC*, pages 146–160, 2008.

BBC News. Jail for Facebook spoof Moroccan. online, 2008. URL `http://news.bbc.co.uk/2/hi/africa/7258950.stm`. [Retrieved 2008-12-01].

BBC News. News Corp in $580m internet buy. online, 2005. URL `http://news.bbc.co.uk/2/hi/business/4695495.stm`. [Retrieved 2009-01-10].

Chris Bizer and Daniel Westphal. Developers guide to semantic web toolkits for different programming languages, 2007. URL `http://www4.wiwiss.fu-berlin.de/bizer/toolkits/\#python`. [Retrieved 2008-10-17].

M. Bolin, P. Rha, and R.C. Miller. Automation and customization of rendered web pages. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 163–172. ACM New York, NY, USA, 2005.

C. Brook. Scraping Gmail with Mechanize and Hpricot. online, 2009. URL `http://schf.uc.org/articles/2007/02/14/scraping-gmail-with-mechanize-and-hpricot`. [Retrieved 2009-01-14].

BusinessWeek. News Corp.'s Place in MySpace. online, 2007. URL `http://www.businessweek.com/technology/content/jul2005/tc20050719_5427_tc119.htm`. [Retrieved 2009-01-14].

R.B. Cialdini. *Influence: science and practice*. Allyn and Bacon, 2001.

R.B. Cialdini. The Science of Persuasion. *Scientific American Mind*, 2004.

CNET News. ComScore: Facebook is beating MySpace worldwide. online, 2008. URL http://news.cnet.com/8301-13577_3-9973826-36.html. [Retrieved 2009-01-03].

Comscore. Social Networking Goes Global. online, 2008. URL http://www.comscore.com/press/release.asp?press=1555. [Retrieved 2008-06-14].

P. Corbett. Facebook demographics and statistics report: 276% growth in 35-54 year old users. online, 2009. URL http://www.istrategylabs.com/2009-facebook-demographics-and-statistics-report-276-growth-in-35-54-year-old-users/. [Retrieved 2009-02-10].

CPAN. Frequently Asked Questions about WWW::Mechanize. online, 2009. URL http://search.cpan.org/dist/WWW-Mechanize/lib/WWW/Mechanize/FAQ.pod\#JavaScript. [Retrieved 2009-02-03].

J. DiMicco, D.R. Millen, W. Geyer, C. Dugan, B. Brownholtz, and M. Muller. Motivations for social networking at work. In *Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 711–720. ACM New York, NY, USA, 2008.

H. Enea, K.M. Colby, and H. Moravec. Idiolectic language-analysis for understanding doctor-patient dialogues. pages 278–284, 1973.

R. Epstein. From Russia, with Love. *Scientific American Mind*, 18(5):16–17, 2007.

Facebook. Public search listings on facebook. online, 2007. URL http://blog.facebook.com/blog.php?post=2963412130. [Retrieved 2008-11-17].

Facebook. Welcome to Facebook, everyone. online, 2006. URL http://blog.facebook.com/blog.php?post=2210227130. [Retrieved 2008-12-28].

Facebook. Facebook - help center. online, 2009a. URL http://www.facebook.com/help.php?hq=disabled&ref=hq#. [Retrieved 2009-02-15].

Facebook. Facebook statistics. online, 2009b. URL http://www.facebook.com/press/info.php?statistics. [Retrieved 2009-02-18].

Facebook. Facebook - terms of use (date of last revision: September 23, 2008). online, 2009c. URL http://www.facebook.com/home.php#/terms.php?ref=pf. [Retrieved 2009-02-19].

FacebookBots.com. Facebook bot reviews. online, 2009. URL http://www.facebookbots.com/. [Retrieved 2009-02-15].

A.J. Ferguson. Fostering e-mail security awareness: The West Point carronade. *Educause Quarterly*, 28(1):54–57, 2005.

P. Finn and M. Jakobsson. Designing and Conducting Phishing Experiments. *IEEE Technology and Society Magazine, Special Issue on Usability and Security*, 26(1): 46–58, 2007.

B. J. Fogg and Daisuke Iizawa. Online Persuasion in Facebook and Mixi: A Cross-Cultural Comparison. In *PERSUASIVE*, pages 35–46, 2008.

Hagen Fritsch. StudiVZ - Inoffizielle Statistiken vom Dezember 2006. online, 2008. URL `http://studivz.irgendwo.org/`. [Retrieved 2008-11-29].

Gartner Inc. There Are No Secrets: Social Engineering and Privacy. *Garnter Security webletter*, 1(1), February 2002a. URL `http://www.gartner.com/gc/webletter/security/issue1/`. [Retrieved 2008-10-29].

Gartner Inc. Protect Against Social Engineering Attacks. *Garnter Security webletter*, 1(1), February 2002b. URL `http://www.gartner.com/gc/webletter/security/issue1/article2.html`. [Retrieved 2008-11-13].

Robert Gibson. Who's really in your top 8: network security in the age of social networking. In *SIGUCCS '07: Proceedings of the 35th annual ACM SIGUCCS conference on User services*, pages 131–134, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-634-9. doi: http://doi.acm.org/10.1145/1294046.1294077.

R. Gold. Frequently Asked Questions about WWW::Mechanize. online, 2009. URL `{http://httpunit.sourceforge.net/doc/javascript-support.html}`. [Retrieved 2009-02-05].

S. Golder, D.M. Wilkinson, and B.A. Huberman. Rhythms of social interaction: messaging within a massive online network. *Arxiv preprint cs.CY/0611137*, 2006.

D. Gragg. A Multi-Level Defense Against Social Engineering. *SANS Reading Room, March*, 13, 2003.

S. Granger. Social Engineering Fundamentals, Part I: Hacker Tactics. *SecurityFocus*, 2001. URL `http://www.securityfocus.com/infocus/1527`. [Retrieved 2008-10-02].

S. Granger. Social Engineering Fundamentals, Part II: Combat Strategies. *SecurityFocus*, 2002. URL `http://www.securityfocus.com/infocus/1533`. [Retrieved 2008-12-23].

S. Grazioli. Where Did They Go Wrong? An Analysis of the Failure of Knowledgeable Internet Consumers to Detect Deception Over the Internet. *Group Decision and Negotiation*, 13(2):149–172, 2004.

R. Gross and A. Acquisti. Information revelation and privacy in online social networks (the Facebook case). In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, 2005.

E. Hargittai. Whose Space? Differences Among Users and Non-Users of Social Network Sites. *Journal of Computer-Mediated Communication*, 13(1):276–297, 2007.

C. Herley and D. Florencio. Phishing as a Tragedy of the Commons. *NSPW 2008, Lake Tahoe, CA*, 2008.

M. Hermansson and R. Ravne. Fighting Social Engineering. Master's thesis, 2005. URL `http://www.dsv.su.se/research/seclab/pages/pdf-files/2005-x-281.pdf`.

G. Hogben. Security Issues and Recommendations for Online Social Networks. *Position Paper. ENISA, European Network and Information Security Agency*, 2007.

D. Huynh, S. Mazzocchi, and D. Karger. Piggy Bank: Experience the Semantic Web inside your web browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):16–27, 2007.

iOpus Inc. imacros feature comparison - free and business editions. online, 2009. URL `http://www.iopus.com/imacros/compare/all/`. [Retrieved 2009-02-12].

T.N. Jagatic, N.A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

H. Jones and J.H. Soltren. Facebook: Threats to Privacy. *Project MAC: MIT Project on Mathematics and Computing*, 2005.

E.A. Kolek and D. Saunders. Online Disclosure: An Empirical Examination of Undergraduate Facebook Profiles. *NASPA Journal*, 45(1), 2008.

S. Kowalski. *IT Insecurity: A Multi-disciplinary Inquiry*. PhD thesis, University of Stockholm and Royal Institute of Technology, Stockholm, Sweden, 1994.

S. Kowalski, S. Walentowicz, and A.R. Mozuraite. Using Chatbots for Security Training of ICT Users. Wireless Word Research Forum 20th Conference, 2008.

B. Krulwich. Automating the Internet: Agents as User Surrogates. 1997.

R. Levine. *Power of persuasion*. John Wiley & Sons, 2003.

H.G. Loebner. Loebner Prize in Artificial Intelligence. online, 2009. URL `{http://www.loebner.net/Prizef/loebner-prize.html}`. [Retrieved 2009-02-12].

F. Manola, E. Miller, and B. McBride. RDF Primer. W3C Recommendation 10 February 2004. *Consultado en*, pages 31–8, 2006.

K. Marett, D.P. Biros, and M.L. Knode. Self-efficacy, Training Effectiveness, and Deception Detection: A Longitudinal Study of Lie Detection Training. *lecture notes in computer science*, 3073:187–200, 2004.

Microsoft. Facebook and Microsoft Expand Strategic Alliance. online, 2007. URL http://www.microsoft.com/Presspass/press/2007/oct07/10-24FacebookPR.mspx2007. [Retrieved 2009-01-12].

K.D. Mitnick and W.L. Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.

T. Muller. 13 reasons your facebook account will be disabled. online, 2008. URL http://getsatisfaction.com/facebook/topics/13_reasons_your_facebook_account_will_be_disabled. [Retrieved 2009-02-13].

K. Näckros. Learning Security through Computer Games: Studying user behaviour in a real-world situation. *International Federation for Information Processing - IFIP*, 237:95, 2007.

R. Nelson. Methods of Hacking: Social Engineering. online, 2008. URL http://www.isr.umd.edu/gemstone/infosec/ver2/papers/socialeng.html. [Retrieved 2008-10-20].

Net Applications. Top Browser Share Trend. online, 2009. URL http://marketshare.hitslink.com/report.aspx?qprid=1. [Retrieved 2009-01-20].

M. Nohlberg. Why Humans are the Weakest Link. *Social and Human Elements of Information Security: Emerging Trends and Countermeasures*, page 15, 2008.

M. Nohlberg and S. Kowalski. The Cycle of Deception-A Model of Social Engineering Attacks, Defences and Victims. In *Proceedings of the Second International Symposium on Human Aspects of Information Security & Assurance (HAISA 2008)*, July 2008.

M. Nohlberg, S. Kowalski, and M. Huber. Measuring Readiness for Automated Social Engineering. In *In CD ROM Proceedings of the 7th Security Conference*, pages 20.1–20.13., June 2008. ISBN 978-1-935160-01-4.

T. O'Reilly. What Is Web 2.0-Design Patterns and Business Models for the Next Generation of Software, 2005. online, 2008. URL http://web.archive.org/web/20071128121019/http://www.oreillynet.com/lpt/a/6228. [Retrieved 2008-10-01].

D. Pagkalos. New highly critical facebook xss vulnerabilities pose serious privacy risks. online, 2008. URL http://www.xssed.com/news/80/New_highly_critical_Facebook_XSS_vulnerabilities_pose_serious_privacy_risks/. [Retrieved 2009-01-10].

Python Software Foundation. General python faq - 1.3 what is python good for? online, 2009. URL http://www.python.org/doc/faq/general/\#what-is-python-good-for. [Retrieved 2009-02-15].

T. Qin and J.K. Burgoon. An Investigation of Heuristics of Human Judgment in Detecting Deception and Potential Implications in Countering Social Engineering. *Intelligence and Security Informatics, 2007 IEEE*, pages 152–159, 2007.

The Register. Chocolate the key to uncovering PC passwords. online, 2008. URL `http://www.theregister.co.uk/2007/04/17/chocolate_password_survey/`. [Retrieved 2008-10-14].

J. Robertson. Scams use striptease to break web traps. online, 2007. URL `http://www.sfgate.com/cgi-bin/article.cgi?f=/n/a/2007/10/31/financial/f163115D11.DTL`. [Retrieved 2009-02-10].

D. Rosenblum. What Anyone Can Know: The Privacy Risks of Social Networking Sites. *Security & Privacy, IEEE*, 5(3):40–49, May-June 2007. ISSN 1540-7993. doi: 10.1109/MSP.2007.75.

Sophos. Facebook - profile privacy settings - sophos recommends. online, 2009. URL `http://www.sophos.com/security/best-practice/facebook.html`. [Retrieved 2009-01-23].

SPAMfighter. Facebook receives $873 million compensation from spammer adam guerbuez. online, 2008. URL `http://www.spamfighter.com/News-11429-Facebook-Receives-$873-Million-Compensation-from-Spammer-Adam-Guerbuez.htm`. [Retrieved 2009-02-20].

Spokeo. Spokeo/hr - Explore beyond the resume. online, 2008. URL `http://www.spokeo.com/hr`. [Retrieved 2008-12-01].

S. Srikwan. Using Cartoons to Teach Internet Security. *Cryptologia*, 32(2):137–154, 2008.

S. Stasiukonis. Social Engineering, the USB Way. *DarkReading*, 2006. URL `http://www.darkreading.com/security/perimeter/showArticle.jhtml?articleID=208803634`. [Retrieved 2008-10-29].

Suburban Journals. 'My Space' hoax ends with suicide of Dardenne Prairie teen. online, 2008. URL `http://suburbanjournals.stltoday.com/articles/2007/11/11/news/sj2tn20071110-1111stc_pokin_1.ii1.txt`. [Retrieved 2008-11-28].

The Guardian. The oldest profession combined with the newest technology. online, 2007. URL `http://www.guardian.co.uk/technology/2007/dec/13/internet.crime`. [Retrieved 2009-02-17].

Chicago Tribune. Facebook child porn case results in 35-year federal prison term. online, 2008. URL `http://archives.chicagotribune.com/2008/oct/22/local/chi-fox-lake-facebook-both-22-oct22`. [Retrieved 2008-11-29].

A.M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

J. Vascellaro. Social networking goes professional. *Wall Street Journal*, 2007.

Richard S. Wallace. The Annotated A.L.I.C.E. AIML. online, 2009. URL `http://www.alicebot.org/aiml/aaa/`. [Retrieved 2009-02-07].

Richard S. Wallace. Don't Read Me - A. L. I. C. E. and AIML Documentation. online, 2000. URL `http://alicebot.org/articles/wallace/dont.html`. [Retrieved 2009-02-10].

J. Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1): 36–45, 1966.

Wikipedia. Social network service. online, 2008a. URL `http://en.wikipedia.org/wiki/Social_network_service`. [Retrieved 2008-10-20].

Wikipedia. List of social networking websites. online, 2008b. URL `http://en.wikipedia.org/wiki/List_of_social_networking_websites`. [Retrieved 2008-10-10].

Wikipedia. Cross-site scripting. online, 2008c. URL `http://en.wikipedia.org/wiki/Cross-site_scripting`. [Retrieved 2008-10-15].

Wikipedia. Stalking. online, 2008d. URL `http://en.wikipedia.org/wiki/Stalking`. [Retrieved 2008-11-23].

Wikipedia. Phishing. online, 2009. URL `http://en.wikipedia.org/wiki/Phishing`. [Retrieved 2009-01-04].

L. Yngström. *A Systemic-Holistic Approach to Academic Programmes in IT Security*. PhD thesis, University of Stockholm and Royal Institute of Technology, Stockholm, Sweden, 1996.

Part IV

Appendix

Appendix A

Chickenscratch/Javascript classes

This version of the thesis doesn't include the source code of the ASE bot. Please contact the author (mhuber@dsv.su.se) if you would like to get a copy of the source code.

Appendix B

Phyton classes

This version of the thesis doesn't include the source code of the ASE bot. Please contact the author (mhuber@dsv.su.se) if you would like to get a copy of the source code.

Appendix C

AIML configuration

This version of the thesis doesn't include the source code of the ASE bot. Please contact the author (mhuber@dsv.su.se) if you would like to get a copy of the source code.