<table>
<thead>
<tr>
<th>Document type</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>D9.1 – First iteration of software prototypes and report describing those prototypes</td>
</tr>
<tr>
<td>Deliverable Number</td>
<td>D9.1</td>
</tr>
<tr>
<td>Editor(s)</td>
<td>E. Vanderhoven, T. Schellens, M. Valcke, OWK, Ghent University</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>External</td>
</tr>
<tr>
<td>Preparation date</td>
<td>21 December 2012</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Legal Notice
All information included in this document is subject to change without notice. The Members of the IWT SBO SPION project make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the IWT SBO SPION project shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
The IWT SBO SPION Project

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Participant name</th>
<th>Country</th>
<th>Department</th>
<th>Participant role</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KU Leuven</td>
<td>BE</td>
<td>COSIC/ESAT</td>
<td>Coordinator</td>
</tr>
<tr>
<td>2</td>
<td>KU Leuven</td>
<td>BE</td>
<td>DISTRINET</td>
<td>Partner</td>
</tr>
<tr>
<td>3</td>
<td>KU Leuven</td>
<td>BE</td>
<td>DTAI</td>
<td>Partner</td>
</tr>
<tr>
<td>4</td>
<td>KU Leuven</td>
<td>BE</td>
<td>ICRI</td>
<td>Partner</td>
</tr>
<tr>
<td>5</td>
<td>Vrije Universiteit Brussel</td>
<td>BE</td>
<td>SMIT</td>
<td>Partner</td>
</tr>
<tr>
<td>6</td>
<td>University of Ghent</td>
<td>BE</td>
<td>OWK</td>
<td>Partner</td>
</tr>
<tr>
<td>7</td>
<td>Carnegie Melon University</td>
<td>USA</td>
<td>Heinz</td>
<td>Partner</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Ellen Vanderhoven</td>
<td>Ughent, OWK</td>
</tr>
<tr>
<td>2 Tammy Schellens</td>
<td>Ughent, OWK</td>
</tr>
<tr>
<td>3 Martin Valcke</td>
<td>Ughent, OWK</td>
</tr>
<tr>
<td>4 Willem De Groef</td>
<td>KU Leuven, iMinds-DistriNet</td>
</tr>
<tr>
<td>5 Dave Clarke</td>
<td>KU Leuven, iMinds-DistriNet</td>
</tr>
<tr>
<td>6 Frank Piessens</td>
<td>KU Leuven, iMinds-DistriNet</td>
</tr>
<tr>
<td>7 Rula Sayaf</td>
<td>KU Leuven, iMinds-DistriNet</td>
</tr>
<tr>
<td>8 Bo Gao</td>
<td>KU Leuven, DtaI</td>
</tr>
<tr>
<td>9 Bettina Berendt</td>
<td>KU Leuven, DtaI</td>
</tr>
<tr>
<td>10 Ero Balsa</td>
<td>KU Leuven, COSIC</td>
</tr>
<tr>
<td>11 Claudia Diaz</td>
<td>KU Leuven, COSIC</td>
</tr>
<tr>
<td>12 Seda Gürses</td>
<td>KU Leuven, COSIC</td>
</tr>
<tr>
<td>13 Bart Preneel</td>
<td>KU Leuven, COSIC</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Content Table</td>
<td>3</td>
</tr>
<tr>
<td>Abstract</td>
<td>4</td>
</tr>
<tr>
<td>1 General Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2 FlowFox: a web browser to enforce information flow policies on web scripts</td>
<td>6</td>
</tr>
<tr>
<td>3 Scramble! A software tool for enforcing confidentiality and integrity on social networking sites</td>
<td>9</td>
</tr>
<tr>
<td>4 Freebu: a software tool for obtaining feedback from the system for privacy awareness</td>
<td>15</td>
</tr>
<tr>
<td>5 Overall conclusion</td>
<td>19</td>
</tr>
<tr>
<td>References</td>
<td>20</td>
</tr>
</tbody>
</table>
Abstract

This report describes three software prototypes that were developed by the technical partners of the SPION-project, in close collaboration with the user group. Each tool resolves specific privacy- and security problems that occur when using social network sites (SNS). The first tool was developed by iMinds-DistriNet and is a web browser nicknamed FlowFox that prevents online web scripts from leaking sensitive or confidential information, according to a given user-defined policy. The second tool, i.e. Scramble, was developed by COSIC and focuses on keeping the content users share in SNS confidential, regardless of which privacy settings are available to the user in the SNS, by providing the users a means to encrypt the content (e.g., messages, posts, comments) uploaded to the SNS. Finally, the third tool FreeBu was developed by DTAI. It aims to help users in recognizing the grouping structure of their online contacts, as well as in modifying these groups. Hereby, the tools try to offer a solution for the problem of context collision on SNS.
1 General introduction

The SPION-project aims to develop concrete valorization outcomes that are developed and disseminated in close cooperation with the members of the user groups. Based on our discussions with the user group, we have identified the need for better tools for security and privacy protection. Current social networking technologies lack usable mechanisms to ensure security and privacy protection. Therefore, the project aims to provide social network site-providers and developers with software tools to support the privacy-friendly privacy policies and privacy-friendly default settings, that will be described in the project deliverable 9.3. The open-source software tools developed by the technological partners of the project will address some of the existing privacy- and security concerns.

More specifically, three tools have been developed. The first tool is FlowFox [1], the first fully functional web browser that implements a precise and general information flow control mechanism for web scripts, developed by iMinds-DistriNet. It is based on the technique of secure multi-execution that support powerful, yet precise policies refining the same-origin-policy in a way that is compatible with both online social network sites (SNS) and regular web sites. FlowFox is also a fundamental build stone of the privacy-enhanced social application platform PESAP [4].

A second tool is a software plug-in to enable users to encrypt information and obtain confidentiality properties, developed by COSIC. This tool aims to keep the content users share in SNS confidential, regardless of which privacy setting the service provider puts at the users’ disposal in the SNS. Several SNS allow users to decide whether they want to share some information either with their friends, family, coworkers or any other list of contacts. Other SNS have a much more limited set of privacy settings, so that the users must decide whether they want the information to be completely public or private but still accessible to all their contacts. Furthermore, users must trust the SNS providers themselves and allow them to have access to their data. “Scramble!” the tool proposed, provides users with a means to enforce their own privacy settings regardless of which SNS they are using and the privacy settings currently available in the SNS. The tool encrypts the content (e.g., messages, posts, comments) uploaded by the users to the SNS for a certain set of people previously selected by the users, so that nobody else (even the service provider) is able to decrypt the uploaded content and therefore be able to access it. As a result, Scramble enforces confidentiality and integrity for each of piece of data uploaded to the SNS. Moreover, the tool performs encryption and decryption as well as integrity checks automatically, meaning that users never have to deal with complex cryptographic techniques themselves.

The third tool is called FreeBu and is developed by DTAI. It is a tool for obtaining feedback from the system to raise privacy awareness. Indeed, in SNS it can be difficult to maintain the context of a conversation or action, i.e. to know what the situation is and how to act appropriately. The resulting uncertainties may lead to privacy issues. In developing the tool, there was a focus on one issue,
i.e. context collision. We propose that a first step to address this issue is to help users distinguish groups of contacts within their SNS-accounts. Therefore, a small user study was conducted, to investigate the criteria of users grouping the people they know. We summarized our participants’ strategies of labeling the groups and found that they based the grouping mainly on their connections with others. We used these results in the design of FreeBu, a semi-automatic and interactive grouping tool, which is based on mining friend graph data for community detection and profile information for labeling.

In what follows, all tools are described more extensively. It is mentioned how different partners of the project cooperated to create tools that are adapted to the target users. A detailed description of the tool, together with a step-by-step guide is given for every tool. Finally, it is concluded how these tools contribute to the existing set of software tools.

2 FlowFox: a web browser to enforce information flow policies on web scripts

2.1 Introduction

An important contributor to the success of social networks is their support for third-party applications. Such applications provide social-enhanced features or functionality (e.g. quizzes) or social games. With such applications, the code provider is typically a third stakeholder, next to the social network provider and the end users. Since these applications are social-aware, they need access to private information of the user to reach their full potential. As a consequence, there is a strong need for security and privacy controls, and enforcing such controls is complex because of the many stakeholders involved.

In the context of web-based online social networks (the majority of current social networks are web-based), third-party applications are typically developed in scripting languages like JavaScript, and access control or information flow control is addressed at the level of the scripting language. These third-party applications, like all web scripts, are typically a combination of markup and executable scripts where the scripts can interact with their environment through a collection of powerful APIs that offer communication to remote servers, communication with other pages displayed in the browser, and access to user, browser and application information including information such as the geographical location, clipboard content, browser version and application page structure and content. With the advent of the HTML5 standards, the collection of APIs available to scripts has substantially expanded.

An important consequence is that scripts can be used to attack the confidentiality or integrity of that information. Scripts can leak session identifiers, inject requests into an ongoing session, sniff the user’s browsing history, or track the user’s behavior on a web site. Such malicious scripts can enter a web page because of a cross-site scripting vulnerability, or because the page integrates third party scripts such as advertisements, or gadgets. A recent study has shown that
almost all popular web sites include such remotely-hosted scripts [3]. The importance of these attacks has led to many countermeasures being implemented in browsers. The first line of defense is the same-origin-policy (SOP) that imposes restrictions on the way in which scripts and data from different origins can interact. However, the SOP is known to have holes [5], and all of the attacks cited above bypass the SOP. Hence, additional countermeasures have been implemented or proposed. Some of these are ad-hoc security checks added to the browser (e.g. to defend against history-sniffing attacks, browsers responded with prohibiting access to the computed style of HTML elements [8]), others are elaborate and well thought-out research proposals to address specific subclasses of such attacks (e.g. AdJail [6] proposes an architecture to contain advertisement scripts).

Several researchers have proposed information flow control as a general and powerful security enforcement mechanism that can address many of these attacks, and hence reduce the need for ad-hoc or purpose-specific countermeasures. Several prototypes that implement some limited form of information flow control have been developed. However, general, flexible, sound and precise information flow control is difficult to achieve, and so far nobody has been able to demonstrate a fully functional browser that enforces sound and precise information flow control for web scripts. The tool developed by iMinds-DistriNet, i.e. FlowFox, is the first available web browser that can enforce general information flow security, based on confidentiality policies on the interactions between web scripts and the browser.

This tool has been developed in close collaboration with the other partners of the SPION-project and the users. First, the frequent SPION meetings were the ideal place for discussions amongst the different partners. The outcome of these meetings has highly influenced our research. Also the SPION technical workshop and the numerous discussions with the participants of the workshop, contributed to the overall quality of our work. Second, we’ve had intense contact with Netlog, one of our user group partners to discuss several privacy issues concerning social application platforms. These discussions have influenced our research in general and more particular our work on privacy-enhanced application platforms [4].

2.2 Description of the tool

FlowFox is a fully functional web browser enhanced with an information flow technology named secure multi-execution [2]. FlowFox is implemented on top of Mozilla Firefox 8.0.1. In practice, FlowFox prevents web scripts from leaking sensitive or confidential information, according to a given user-defined policy. FlowFox can also be applied in the context of an innovative privacy-enhanced social application platform (PESAP) by preventing de-anonymized information to leave the browser towards social application platforms.

The installation procedure of FlowFox is the same as for an original Mozilla Firefox. Depending on the used operating system, the procedure, however,
can vary.\textsuperscript{1} The project home page (https://distrinet.cs.kuleuven.be/software/FlowFox/.) contains detailed instructions on how to download & install FlowFox. The main installation idea on all operating systems is to (1) download the installation package, (2) install the package on your local machine and (3) run FlowFox.

To download the installation package, visit the FlowFox download page in any browser. On the download page, you will find a link to the latest FlowFox version. Depending on your connection speed, the download may take up to a few minutes. After downloading, you have to unpack the installation package, according the explanation on the download page. After the first step, the user ends up with a directory containing all necessary files. The user can decide to install FlowFox system-wide, however this is not required.

At start-up, the user needs to specify a policy file. This policy file contains the policies that will be enforced by FlowFox during the current session. The installation package comes with a pre-configured policy file. While surfing the web, the user-defined policies are hidden for the user and are meant for expert users only. An explanation on how to modify a confidentiality policy for FlowFox, is described in detail on the project website.

Once the user starts surfing on the web, each piece of JavaScript will be executed under the secure multi-execution regime, in correspondence with the given policy. If such a web script doesn’t leak sensitive information, the observable program semantics will not change. However, when a web script tries to leak sensitive information, FlowFox will fix the leak while trying to preserve the original functionality of the script as much as possible.

The following example shows how this process works. Imagine the scenario of a malicious advertisement script, embedded in an online e-mail application. The original behavior of this script would be twofold: to leak the content of the user’s inbox and to show a banner image. However, if this scenario would be executed within FlowFox, the user would still see some banner image — because the functionality is preserved as much as possible —, but the leak will be closed.

## 2.3 Conclusion

Several researchers have proposed information flow control as a general and powerful security enforcement mechanism that can address many web-script attacks, and hence reduce the need for ad-hoc or purpose-specific countermeasures. Several prototypes that implement some limited form of information flow control have been developed. However, general, flexible, sound and precise information flow control for online web applications is difficult to achieve, and so far nobody has been able to demonstrate a fully functional browser that enforces sound and precise information flow control for web scripts. As a consequence, there was no evidence for the practicality of this approach in the context of web applications in general and social web applications in particular, up until now. We developed FlowFox, the first fully functional web browser that implements a precise and

\textsuperscript{1}We currently only provide a build of FlowFox that requires Ubuntu Linux (32bit).
general information flow control mechanism based on the technique of secure multi-execution. In practice, FlowFox prevents any web scripts from leaking sensitive or confidential information, according to a given user-defined policy.

In our main research paper [1], we have discussed the design, implementation and evaluation of FlowFox, a browser that extends Mozilla Firefox with a general, flexible and sound information flow control mechanism. FlowFox provides evidence that information flow control can be implemented in a full-scale web browser, and that doing so, supports powerful security policies without compromising compatibility. In our second paper [4], we presented a framework, that includes FlowFox, for a privacy enhanced social application platform (PESAP), that technically enforces the protection of the personal information of a user, when interacting with social applications.

3 Scramble! A software tool for enforcing confidentiality and integrity on social networking sites

3.1 Introduction

Current social networking sites (SNS) provide a range of privacy settings for users to configure and decide who can access the content they upload to the site. Several issues arise from this situation. To begin with, the privacy settings are unilaterally designed and enforced by the service provider. This means that the user has a limited set of choices regarding who is able or unable to access the content he/she uploads to the site. For example, the service provider may provide the user with few choices: either all data will be public to all users on the Internet, either just to his/her full list of friends or either just to his/herself. Hence, this limited set of choices may be in conflict with the user's desire to share certain items of information with just a subset of his/her friends, with this set being different for certain items of information and dynamic in time. The lack of flexible and easily tunable privacy settings may cause users to perceive their privacy as being violated and even force them to refrain from uploading content to the SNS.

Because the privacy settings are unilaterally enforced by the service provider, the provider may as well decide to change them at any given time, even after giving no notice. Changes in privacy settings directly affect the visibility of the content users upload to SNS, making it available to a wider or shorter audience than the user initially intended. Both the widening and narrowing of the availability of users' content may affect seriously the users' privacy. On the one hand, making the users' data available to a wider audience potentially discloses sensitive information to parties the users might rely on not being disclosed to, e.g., a user organizing a surprise party for a friend relies on information about this party not being available to this friend. On the other hand, making the users' data available to a narrower audience potentially prevents users poten-
tially expected by the user from accessing that information, e.g., information about the surprise party is not available to all the invited guests anymore.

Lastly, regardless of how coarse or fine grained the privacy settings designed by the service provider are, the provider is able to access all the users’ data. This means that all the activities performed on and the contents uploaded to the online social network are being monitored and processed by a single, all-powerful entity. Users unwilling to disclose their information to the service provider have no choice but to refrain using the service. All other users may face several privacy threats resulting from the collection of highly sensitive data by the service provider: from insider attacks by the provider employees to the pressure of the police or any other law enforcement authorities, from the data mining performed by marketers and the pressure of companies to gather accurate information about consumer’s choices and behaviours, the service provider may leak intentionally or unintentionally very sensitive and personal information from the users.

Scramble!, the tool we present in this document, aims at solving all three problems aforementioned, by means of encryption. This provides an alternative for the user to decide who is able to access the data she uploads on the SNS, so that no unauthorised user, including the service provider, is able to access his/her data, regardless of the site’s privacy settings and further changes on these settings. Previous works such as Lockr [16, 15], NOYB [17] or FaceCloak [18] have provided a similar solution to these problems, namely, a solution based on cryptographic techniques implemented either as a browser plug-in or a Facebook application. The main difference between these solutions and Scramble! is that whereas the former are platform specific, Scramble! is SNS-independent, i.e., it can be used in any site.

Scramble! has been developed in close collaboration with the other partners of the SPION-project. This collaboration is varied and happens at different frequency, intensity and levels. First of all, our frequent SPION meetings are the perfect opportunity to discuss the details and intricacies of our research and the development of the valorization tool. In these meetings we obtain valuable feedback for better designing and improving our tool. Moreover, through the joint examination of the conceptual framework inherent to the SPION project we have come across several issues that are important for the successful development of our tool.

Indeed, iMinds-SMIT has arisen our research interest on several privacy issues, such as context collision[10], context collapse[7], invisible audiences [14]and the blur between public and private[7], to which our tool has managed to provide a solution. Moreover, iMinds-SMIT has mentioned serious privacy problems stemming from the unconsciousness of what is happening with the users’ data online. Because our tool uses robust encryption techniques, any leaked data are encrypted thus no unauthorised party can access to their content. Finally, iMinds-SMIT has highlighted the importance of developing tools that while tackling the raised privacy issues do not imperil the positive features represented by SNS.

Interaction with OWK helps us developing a more pedagogical language to-
wards communicating with users and understanding their needs. This is important in order to be able to include their requirements in the developing process, as targeting the privacy problems users perceive is the essence of this tool. Furthermore, interaction with iMinds-DistriNet and DTAI (the two other technical partners in SPION) has provided us valuable feedback regarding how to address other privacy problems from a technical perspective, as well as highlighting the core problems to users’ privacy in online social networks.

Finally, we are starting to closely collaborate with CMU. We have started conversations about the evaluation of the tool from the economics of privacy and behavioural economics points of view. Guided by the expertise of CMU in those fields, we are planning to execute a thorough evaluation. The result of such evaluation should shed light on several questions related to understanding the behaviour of the users of the tool. Understanding how the privacy perception of users changes when using the tool, whether or not their behaviour changes as they are given more control over the disclosure of their data and to what extent the tool meets their expectations of privacy protection are amongst several questions that we want to answer with this evaluation. The results of this evaluation will be reported in SPION-deliverable 9.3 (Report containing “usability and behavioral evaluations of the software tools”).

3.2 Description of the tool

“Scramble!”, is a client side application implemented as a browser plug-in (a Firefox extension) that users can install on their computers to keep their SNS data confidential. Scramble! generates cryptographic keys\(^2\) and automatically encrypts the content users post in SNS so that only the friends they choose to share it with are able to decrypt it. Note that this requires other users to install Scramble as well, i.e., a user must install Scramble to be able to decrypt the messages sent or posted by their friends. For example, if Alice uses Scramble! to send encrypted messages to her friends Bob and Charlie, they need to install Scramble to be able to read Alice’s messages\(^3\).

Scramble therefore guarantees the confidentiality of the users data towards any user or entity, such as the service provider, that the users do not grant access to. Scramble contains an easy-to-use user interface for defining the set of users the data should be shared with.

“Scramble!” provides a solution to users of social networking sites that either want to conceal the content of the data they upload to the service provider, either are unsatisfied with the privacy settings enforced by the social network provider, or both.

After the users install “Scramble!” on Firefox, a small Rubik’s cube icon will

---

\(^2\)The public key pairs can also be provided by the user.

\(^3\)Strictly speaking, this is not necessarily true. Bob and Charlie could manually generate their own public key pairs, share them with Alice and then decrypt the messages themselves. However, this would require Bob and Charlie being able to deal with cryptographic mechanisms. What Scramble! precisely does is to manage complex cryptographic operations in the background.
appear at the right bottom of the browser window. This indicates that Scramble was successfully installed. Right after installation, users can either ask Scramble! to generate cryptographic keys for them or use their own cryptographic keys. After generating the keys, Scramble! automatically uploads the public keys to a server on the Internet so that the users’ friends are able to retrieve them, while keeping the private key securely on each user’s computer. In fact, to obtain the keys of their friends, users can either provide to Scramble! their users’ friends e-mail addresses or also provide them directly to Scramble! from a text file, for example.

After the key generation process Scramble! is ready to use. In order to do that, users just write messages in the fields provided in the SNS, as they would normally. The only difference while using Scramble is that, before posting/sending the message, the user selects the message\(^4\) of which he/she wants to restrict availability to a certain subset of his/her contacts. Then, the user right-clicks on the selected message to obtain the dialogue that leads to the Scramble tool. Figure 1 shows this process. The user selects Scramble → Scramble It! and is directed to a new window where he/she can select the people he/she wants to be able to decrypt the message.

![Figure 1: Selecting a message to be encrypted with Scramble!](image)

Alternatively, instead of right-clicking the message, the user may click on the small Rubik’s cube at the right bottom of the browser window, where the same option “Scramble It!” is available. This is shown in Figure 2.

Once the option “Scramble It!” is selected, a new different Scramble! window is shown by the browser, as shown in Figure 3. In this window, the user is able

\(^4\)or part of the message
Figure 2: Using the small rubik’s cube icon in the browser to access Scramble!

to select the contacts he/she wants to share the message with. He/she can either select full groups or choose person by person, whatever the needs are. Once the user selects the people wanted to access the message, one clicks done, and the window will be closed bringing the user back to the SNS.

Scramble had encrypted by now the message with the special Scramble headers, as shown in Figure 4. The message is now ready to be posted.

Once the message has been posted, Scramble will decrypt the message automatically for the user, as well as for the contacts chosen by the user to be able to decrypt and read the message. Therefore, no user has to deal with the complicated cryptographic techniques used by Scramble. It is easy to use and transparent to the user.

Scramble can be downloaded here: http://sourceforge.net/projects/scramble-it/files/latest/download

Instructions on how to install Scramble can be found here: http://homes.esat.kuleuven.be/feato/extras/scramble/install.html

Instructions on how to use Scramble can be found here: http://homes.esat.kuleuven.be/feato/extras/scramble/howto.html

3.3 Conclusion

We have implemented Scramble, a Firefox extension that provides users with a way of enforcing confidentiality and integrity to the content they upload on social network sites. Scramble is SNS independent thus can be used in various
SNS, such as Twitter, Facebook, Google+ or MySpace. Furthermore, not only is it SNS independent, but it can also be used with other Web 2.0 services, such as blogs, forums and wikis. Potentially, it allows users to store data in encrypted format in any cloud service.

The extension allows the definition of groups to ease the task of selecting which users should be granted access to the user content, as well as the encryption of content under the keys of all group members. Using a public key encryption scheme we are able to protect the integrity and confidentiality of user created data, especially towards the service provider. Scramble is easy to use and transparent for the users, as it automatically decrypts the content for the authorised users.
In addition to the confidentiality and integrity properties previously mentioned, that tackle problems such as unsuitable, cumbersome or changing privacy settings, as well as the pervasive monitoring of the service provider, there is a substantial amount of added value that users can obtain through the use of Scramble. Because Scramble prevents the service provider from accessing the information posted by the users in the SNS, it prevents inferences made by marketers based on user data that could lead to negative price discrimination or intrusive targeted advertisement.

Moreover, Scramble is publicly available and free. Finally, Scramble is open source, meaning that the technology experienced community can tune it and keep improving it to adapt to current non-addressed or future demands of the users.

4 Freebu: a software tool for obtaining feedback from the system for privacy awareness

4.1 Introduction

One privacy issue that has been widely discussed with regard to SNS is Context Collision or Context Collapse [7, 9, 10]. It refers to the situation that due to technology barriers and the large amount of information in SNS, a user fails to recognize the boundaries between different contexts, and thus behaves inappropriately towards others in an online social environment. For instance, a user might post something that is visible to undesired audiences or that is considered undesirable by some audiences. We recognize that a first step to address this issue is to help a user distinguish different groups of people that he/she knows online, so that the information flow towards supposedly different people can be better managed by the user.

Therefore, we developed a tool, i.e. Freebu, that aims to help users resolve context collision by aiding them in recognizing the grouping structure of their online contacts, as well as in modifying these groups. In the development process, first, we conducted a small user study to investigate what criteria users employ to group their friends and other people they know. This provided us with a basis to generate descriptions (labels) of detected communities in the tool.

Second, based on the findings from the user study, we motivated our choice for an approach based on data mining. More specifically, we used a graph-based community detection algorithm to extract groups of the user’s contacts. Third, we described a method and an interactive tool for community detection and labeling.

Freebu provides the user with a semi-automatic grouping solution. It first presents the user with a fully-automated grouping suggestion. As mentioned before, this suggestion is constructed based on our user study to mimic users’ grouping behavior in general, so that the groups look natural or sensible to the user. This saves the user a significant amount of time, because he/she does not need to categorize all the friends into groups manually. Next, if the user feels like
modifying the grouping structure, he/she can drag and drop people to different
groups, create new (sub)groups, or remove certain (sub)groups. Once satisfied
with the grouping structure, users can directly publish the groups as friend lists
on their Facebook account. This means that a group in FreeBu corresponds to
a friend list in the user's Facebook account. With such an approach, the user
can create friend lists effectively and efficiently.

In our process of developing this software tool, we collaborated closely with
the other partners of the SPION-project. We had many meetings with iMinds-
DistriNet, discussing ideas of underlying access control models and top-level
interfaces for assisting SNS users in general to protect their privacy. These
discussions focused on the needed requirements, the definition of contexts and
how these contexts can be used, which became an inspiration for the later tool
development.

iMinds-SMIT’s argument in deliverable 2.1 State Of The Art (SOTA) on the
lack of situation or context in SNS is very relevant to our research. It pointed
out the literature about context collision or context collapse which served as
our direct motivation for building our grouping tool for context management in
SNS. iMinds-SMIT’s focus on the relationship between offline and online com-
unities, especially the part on the close connections between online and offline
environment, led us to taking into account online as well as offline contacts
in the questions we asked during a small user study of grouping behavior, de-
signed to inform tool development. Discussions with iMinds-SMIT also led us
to adopt the method for grouping elicitation that they use: letting users cre-
ate tree structures to externalize their mental groupings. At the moment, we
are collaborating with iMinds-SMIT on an evaluation study of FreeBu, which
will be reported in SPION-deliverable 9.3 (Report containing “Evaluation of
the software tools from the perspective of the user expectations, practices, and
context of use”).

CMU listed several cases of information disclosure of users on Social Net-
working Sites and the privacy issues that stem from such disclosure. CMU then
showed several possible aiding strategies that have been described in the liter-
ature to protect users’ privacy, which include (1) “design approaches in group
context” and (2) “soft paternalism”. The former pointed out the need for the
management of group contexts in SNS, the latter emphasized that without coer-
cion, technology can guide users to make more appropriate decisions concerning
their privacy. Both have contributed to the idea and the building of our grouping
tool.

OWK’s comments on “raising awareness about information flow” and “learn
how to build an online identity” mentioned in the “think before you post” sec-
tion in the SOTA also contributed to the development of our tool. Furth-
more, OWK pointed out that among the current educational packages that
guide young users to protect their privacy in using the internet, especially SNS,
few refer to real technical skills like changing the privacy settings. We consider
our grouping tool with vivid visualizations as a possible aid for countering these
possible problems. However, although it helps directing the users’ attention
to the privacy settings, FreeBu does NOT directly change the user’s privacy
4.2 Description of the tool

The tool we developed is called FreeBu (Friend Tree Bubbles). It is a desktop application that runs on user's Facebook friend graph and friend profile data. The friend graph and profile data are extracted by a token submitter before running the FreeBu tool. The tool is targeting any Facebook user.

We propose an automatically generated grouping based on the user's friend graph on Facebook. This way, we aim to make the users reflect about the grouping of their friends and to increase their awareness of existing friend groups. For the visualization of the result, we adopt the star-tree form to represent the grouping structure. As shown in Figure 5, the nodes of the tree are represented by circles and each pair of parent-child nodes are connected by straight lines. The root of the tree (the blue circle in the middle) represents the user, the red circles represent different communities detected by the algorithm, the leaves (the green circles surrounding the red ones) represent the user's friends on Facebook. We scale the sizes of community circles based on the number of people within each community; a larger size corresponds to more people.

The labels – including the number of people and the common characteristics of these people – are shown on top of the community circles. As an example, if a community contains one person, only the number “1” is shown as a label to indicate the number of people within this community. The user can click on one bubble – a community or a person – to zoom in and concentrate on a particular part of the tree. In this paper, we blurred the labels for privacy reasons. The labels are typically school names, school years and work places. The number in front of the blurred labels indicates the number of people in the corresponding circles. The user can adjust the number of labels shown by sliding the threshold bar.

Initially, we provide the user with one-layer grouping. The user can modify it by adding or removing (sub) groups, as shown in Figure 6. The user can also change the members of the groups by “dragging and dropping” friends from one red circle to another, as shown in Figure 7.

FreeBu and more detailed instructions can be downloaded via this link: https://dl.dropbox.com/u/62772548/FreeBu.zip.

4.3 Conclusion

People within one context often share some common traits or connections that the people from other contexts lack [12, 13]. Therefore, one important factor that can distinguish different contexts are the persons in it [11, 13]. Hence, with FreeBu, we provide a SNS user with an easy way to identify different groups of people among the user's contacts. The user can “fine-tune” the grouping structure according to his/her needs by intuitively dragging and dropping some bubbles (that represent people) into other bubbles (that represent groups),
Figure 5: The overview of the user interface.

Figure 6: The user can add new groups at different levels of the star-tree. “New group” is added at level two, attached to the level-one circle labeled with “41”. “New group 1” is added at level one, directly attached to the “self” circle. The user can also edit the labels of the circles.

Figure 7: On the left, three individuals are initially assigned into three different groups; On the right, the user move the three individuals into one group.
change the group names, and eventually, the user can publish this grouping decision onto his/her Facebook account. This saves the user a significant amount of time by substituting manually constructing friend lists. The user also gains an overview towards his/her online friends, and is likely to post more appropriately to designated friends based on the groups. We believe that distinguishing different groups of people is a very important step towards online context management.

5 Overall conclusion

There have been several gaps in security- and privacy protection for users of SNS. To answer the needs of users, service providers, and developers, three different software-tools were developed. Each tool tried to offer a solution to another security-problem.

First, the current web infrastructure does not provide adequate protection mechanisms against maliciously-behaving web scripts (e.g. malicious advertisements on online social network sites) that try to leak sensitive or confidential information from the user’s browser towards the information-gathering endpoint of an adversary. In order to assist the helpless internet user, iMinds-DistriNet developed FlowFox, a fully functional web browser enhanced with information flow technology. In practice, FlowFox prevents web scripts from leaking sensitive or confidential information, according to a given user-defined policy. FlowFox has also been successfully applied in the context of an innovative privacy-enhanced social application platform by preventing de-anonymized information to leave the browser towards social application platforms.

The second tool, i.e. Scramble!, is developed by COSIC for any SNS user. It can be used to protect the confidentiality and integrity of the users’ messages, posts or any other content uploaded to the SNS by means of encryption. This way, no unauthorised party is able to access or modify those data. It is valuable because it is freely available, open source, easy to use and it can be used on different (SNS) platforms.

The third tool is called FreeBu and is developed by Drai for Facebook users. It can be used to automatically generate friend lists for Facebook users. The user can gain an overview of his/her friends, reflect on this overview and modify the grouping structure. By doing so, the user makes more informed decisions on SNS.

Although it may seem that the use of one of the tools described above is enough to protect your privacy, the following example will show that all tools presented in this deliverable are complementary. Therefore, the SNS-user will be most protected when using all three of the tools.

Imagine a 16-year old girl Alice, who just had the number of her mobile phone changed. Because she wants to communicate her new mobile number to all of her friends, she posts her new phone number on her Facebook profile-page, with her privacy-settings set to ‘friends-only’. The next day, she received a message on her phone from her teacher, she was called by an employer of Facebook for
a customer satisfaction survey and by a phone company to convince her that choosing another provider would decrease her phone costs.

Although this scenario may sound quite unrealistic, all described consequences of posting a phone number on your SNS-profile page are possible, even when using the privacy settings of your provider. Indeed, by posting the number, this information was not only visible for everyone Alice once accepted as a Facebook-friend (including her teacher), but it was also leaked to the service provider (Facebook) and third companies having adds on her profile page (e.g. a phone company).

Now imagine if Alice and her friends had installed FreeBu, Scramble and Flowfox on their computer. First of all, by using FreeBu, she could easily make friend-lists, something she might not have done without FreeBu because of the workload. These friend-lists are enabling her to choose more specifically who she wants to share her phone number with. In choosing to set her privacy-settings, she might then see that she wants to share her number only with her closest friends and family, and not with all her Facebook-friends, including her teacher. It allows her to make a detailed and specific choice before sharing her phone number. Secondly, to prevent Facebook from seeing her phone number and to perpetuate the SNS-privacy settings, she uses Scramble. Instead of only using the provider’s privacy settings, she also encrypts her phone number, so that only her closest friends and family can decrypt the information. However, in this case, the advertising companies can still see her phone number by making a screenshot at the moment she has just typed in the message on her Facebook profile page, but before she is scrambling the information. Therefore, she also uses Flowfox, disabling the advertisers to take this kind of screenshot and preventing the personal information to be leaked.

This example clearly shows how the three tools presented in this deliverable are complementary and building on a safer and more privacy-friendly SNS-environment. This way, the open-source software developed by the technical partners of SPTION address some of the main existing privacy-and security concerns.

References


