Behav Res
DOI 10.3758/s13428-013-0385-3

Build your own social network laboratory with Social Lab: A tool for research in social media

Pablo Garaizar · Ulf-Dietrich Reips

Abstract Social networking has surpassed e-mail and instant messaging as the dominant form of online communication (Meeker, Devitt, & Wu, 2010). Currently, all large social networks are proprietary, making it difficult to impossible for researchers to make changes to such networks for the purpose of study design and access to user-generated data from the networks. To address this issue, the authors have developed and present Social Lab, an Internet-based free and open-source social network software system available from http://www.sociallab.es. Having full availability of navigation and communication data in Social Lab allows researchers to investigate behavior in social media on an individual and group level. Automated artificial users ("bots") are available to the researcher to simulate and stimulate social networking situations. These bots respond dynamically to situations as they unfold. The bots can easily be configured with scripts and can be used to experimentally manipulate social networking situations in Social Lab. Examples for setting up, configuring, and using Social Lab as a tool for research in social media are provided.

Keywords Internet-based research · Internet science · Social media · Social engineering · Social networking sites · Open-source software

Online communities have played a key role in the development of the Internet from its very beginning (Hauben, Hauben, & Truscott, 1997). Over the years, forums and Usenet groups were gradually replaced by more interactive and easy-to-use instant messaging systems and social networking sites (SNSs). Nowadays, they not only have surpassed e-mail as the dominant form of online communication (Meeker, Devitt, & Wu, 2010), but also try to cover all kinds of communication needs, from micro-interactions (i.e., presence or ratings in the form of "Like," "+1," or votes) to real-time videoconferencing.

Many of the communication scenarios that arise around social media are so novel that sometimes the consequences of their use are neglected. There is much work to do in terms of privacy, security, and trust in this field. Acquisti and Gross (2006) found that Facebook users are only mildly concerned about who can gain access to their personal information and how it can be used. Interestingly, they also found that being the publishers of the content shared on the social network prevented users from being concerned about the sensitivity of the information, because they believe that they have some control over its access. The same happens in SNSs like Twitter, which limits its users to short messages. About a quarter of tweets include information about type, time, and location of activities people are engaging in (Humphreys, Krishnamurthy, & Gill, 2010). The recent success of the mobile versions of SNSs fosters ubiquitous online social interactions and with these, also potential threats to anonymity and privacy. Web sites like *Failbook*[1] or *Lamebook*[2] collect and show many of the problematic uses of SNSs.

The study of Internet-related behavior is flourishing, and a new set of methods for Internet-based research is being developed in psychology (Reips & Birnbaum, 2011). There is now a tradition of tools available for such research, many of which were published in *Behavior Research Methods* (*FactorWiz* and *SurveyWiz*, Birnbaum, 2000; *Form Processor*, Göritz & Birnbaum, 2005; *iScience Maps*, Reips & Garaizar, 2011; *Web experiment list*, Reips & Lengler, 2005; *Web Experimental*

P. Garaizar (✉) · U.-D. Reips
Deusto Institute of Technology (DeustoTech), Universidad de Deusto, Avda. Universidades 24, 48007 Bilbao, Spain
e-mail: garaizar@deusto.es

U.-D. Reips
Ikerbasque, Basque Foundation for Science, Bilbao, Spain

[1] http://failblog.cheezburger.com/failbook
[2] http://www.lamebook.com/

Published online: 06 September 2013

Psychology Lab, Reips, 2001; *WEXTOR*, a Web experiment generator, Reips & Neuhaus, 2002; *VAS Generator*, Reips & Funke, 2008; *Dynamic Interviewing Program* and *User Action Tracer*, Stieger & Reips, 2008, 2010). The number of studies conducted via the WWW with such tools appears to have grown almost exponentially since 1995 (Reips & Krantz, 2010).

In the following sections, we will describe Social Lab and how to use it as a user and as a researcher. We will also provide instructions on how to extend its functionality for particular research purposes, how to adapt it to specific scenarios, and how to translate it into other languages. Finally, we will provide an example of Social Lab's potential use as a simulated social network where users can learn about privacy and social media literacy in a practical and safe way.

Social Lab

Currently, all large social networks (e.g., Facebook, Twitter, Google+, LinkedIn) are proprietary, making it difficult to impossible for researchers who are not associated with the owners to access such networks for the purpose of research. For example, it is impossible to make changes to social networks for the purpose of study of, design of, and access to user-generated data. Some data can be accessed via so-called *APIs*.[3] However, developing research tools on the top of social networks' APIs is often risky and short-lived, due to the frequent changes in their Terms of Service (Watters, 2011). Experimental research about social networks has been severely limited. Often, researchers have been left with no other choice than having participants evaluate a set of static pages or screenshots taken from social networks (Buffardi & Campbell, 2008; Reips & Buffardi, 2012). Researchers lack an SNS that allows participants to dynamically interact with presented scenarios. To solve these issues, we developed Social Lab.

Social Lab is a social network software system designed for research; it is available from http://www.sociallab.es. It is nonproprietary, flexible, free, and open: Any portion of it can be adapted to specific needs, and all navigation and communication data are available to the researcher. Figure 1 shows a Social-lab-based Web site at http://demo.sociallab.es.

There are other social software packages available to deploy social networks. However, most of them are offered as pay-per-use services with limited customization features (e.g., Social Engine, Social Go, phpFox, Webscribble). Some are free and open-source software packages, but difficult to adapt to researchers' needs because they either are derivatives of large content management systems (e.g., BuddyPress, JomSocial) or were created for the purpose of e-learning (e.g., Elgg, Mahara). Conversely, Social Lab was developed for researchers, and it uniquely integrates social bot functionality to stimulate and simulate social interactions.

User perspective

From the point of view of a user, Social Lab has many of the features of a social network. It provides a "social sandbox," a bounded and safe place to socialize, play, and experiment—for the sake of science rather than commerce (e.g., ad free). As in other virtual environments, users of Social Lab are expected to quickly build on and be immersed in social interactions. After signing up and providing informed consent, Social Lab's users can update their personal information (i.e., first name, last name, gender, birthday, e-mail, location, academic information), edit their profiles, and update their status in their wall. To interact, they can add comments to their friends' walls, manage friendship requests, send and receive private messages, share pictures and tag friends on them, create fan pages and become fans of extant pages, or view other users' profiles (the privacy settings of these profiles determine the availability of the information stored in them). Therefore, all users within the same Social Lab server are able to interact with each other.

To take part in a Social-Lab-based site, it is necessary to own a user account. Researchers can either enable the sign-up process for the users in the site or create the required accounts on their own and send them to the participants in the study. The next step is to fill the user's profile with basic personal information. Entries in the name, surname, and e-mail fields of the profile are mandatory, because they are needed for the proper operation of the social network. Once a user profile is filled with basic information, all the features of Social Lab become available. A good starting point is to check the inbox of messages. The first incoming message is sent by Social Lab's system account and contains a briefing about the intended use of the site (see Fig. 2; in this case, a social engineering challenge we implemented as part of a privacy wargame using Social Lab; see below). Similarly, if a desired outcome is achieved, Social Lab's system account can be used to send a new message to the user providing further instructions if the research requires them (e.g., such tasks may include experimental manipulations). However, rather than trying to solve tasks, users may simply pursue their social activities on the network.

Social bots

With the aim of providing a stand-alone tool to conduct social networking related studies, Social Lab includes social bots support among its features. Using social bots, researchers are able to design live situations where users and bots can interact and share information.

Although there are other social bot software packages available, most of them are designed for proprietary SNSs (i.e., Facebook, Twitter, Youtube), and not for a self-hosted

[3] Application programming interfaces, public standardized interfaces to provide limited access to Web applications.

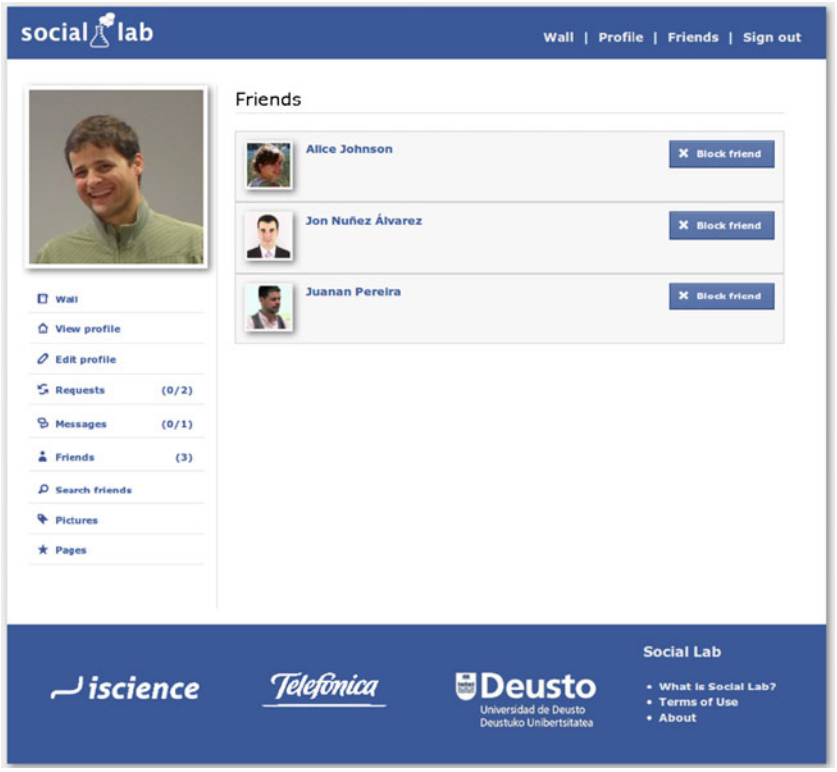

Fig. 1 Social Lab's Web site, user's friends list (English version)

and generic social network. Moreover, considering the lack of integration of these social bots with their target SNSs, changes in APIs or Terms of Services affect them critically. This is not the case with Social Lab's bots. They are fully integrated with the social network and, more interestingly, *stateful*. Stateful bots, in contrast to stateless ones, are able to remember the previous status of an interaction with a user and respond differently according to this status. Even powerful social bot services like IFTTT[4] do not offer stateful interactions. Social Lab allows researchers to set up a bot that will perform the desired outcome (e.g., adding a user as a friend) only if all the previous requirements have been fulfilled (e.g., become a friend of a friend, then send a friendship request talking about music and, finally, a page like a fan page of a rock band).

Each automated behavior in Social Lab is controlled by a social bot. The behavior of these bots is configured using simple scripts that perform checks and actions. Table 1 contains the list of all possible commands that can be used to define the behavior of a social bot (see below for how to expand this list by creating new commands). Table 2 shows the script corresponding to the behavior of "Alice Johnson," the first challenge in our privacy-focused social wargame instance of Social Lab that is explained further in the following section. As can be seen, this bot accepts all incoming friendship requests and, afterward, sends a system message to provide information of the challenge for the next level.

Each time a friendship request is made, Social Lab checks whether it involves a bot, and if that is the case, it schedules a task that corresponds to the script that defines the behavior of the social bot. Periodically, the system checks for scheduled tasks and executes pending steps of each one. As long as the scheduled task dispatcher is not launched too frequently (e.g., every 5 min), the responses of the bots are not immediate, causing interactions similar to those made with real profiles.

In the following, we will provide an example of how researchers can create new bots. No programming knowledge is needed; the backend application discussed below provides a graphical interface for this purpose.

Imagine that a researcher wants to study the SNS users' behavior when they ask to be added as a friend to a profile and they are tagged in a picture immediately after being accepted as a friend by that profile. Several different behaviors can be expected in case users don't agree with the tagging: (1) Users may just untag themselves from the picture, (2) users may decide to block the tagger's profile, (3) users may complain about the tagging behavior via a public or private message, and so forth.

To define a social bot that shows the tagging behavior described above, the following steps need to be taken: (1) Create a user account for the social bot (using the sign-up link at the homepage of Social Lab), (2) edit the user profile and

[4] If This Then That: http://ifttt.com/

Fig. 2 System message providing information about the first challenge of social engineering in Social Lab

upload the picture to the social network, (3) logout from that account and access the backend application using an administrator user account, (4) create a new social bot for Social Lab by adding a new row with the ID of the profile created previously to the *Bot* table (see Fig. 3a), (5) define the behavior of the social bot by adding new rows to the *Step* table in Social Lab's database. To define previously described behavior, *"Accept friendship"* and *"Tag friends in pictures"* commands should be added to the *Step* table (see Fig. 3b, c). As can be seen in Fig. 3, some commands use an extra argument that is a reference to the *Automsg* table. That extra argument is used in commands like "*Send message*," to define which message will be sent to the user, or "*Check request*," to define which text content should be part of the friendship request.

The current set of available commands can be extended with basic knowledge of computer programming. There are three actions required to accomplish this task. First, a new row in the *Command* table should be added, defining the name of the new command in the description field. Second, the identifier of the new command should be defined in the file apps/frontend/config/app.yml. Third, the desired actions or checks should be added using the PHP programming language in the function executeScheduler located in the apps/frontend/modules/default/actions/actions.class.php file. Figure 4 shows the excerpt of code used to create the "Check Friend of a Friend" command, defined in the apps/frontend/config/app.yml file as command_checkfof: 9.

Table 1 Command List (checks and actions) available to define scripts for social bots in Social Lab

Command	Description
1	System message
2	Send message
3	Send wall message
4	Tag friends in pictures
5	Accept friendship
6	Check location match
7	Check academic match
8	Check request
9	Check friend of friend
10	Check friend tagged by player
11	Check page match
12	Check page owner match
13	Check tagged by friend
14	Check friend of two friends
15	Check friend's wall

Data handling

In this section, we describe how events are logged and in which format the data gathered in Social Lab can be accessed by the researcher.

First, Social Lab logs all the interactions within the social network. Each log entry of an interaction contains a corresponding URL. These URLs are written in a format that can be parsed automatically (e.g., the URL of a simple action like editing a user's profile is /profile/edit, and the URL of more complex actions like untagging a user from a picture is also straightforward: /pic/untag/id/ID_PIC/user/ID_USER). Due to the log file format, the use of log file tools such as Scientific LogAnalyzer (Reips & Stieger, 2004) can facilitate the processing of user behavior in Social Lab.

Second, the backend application in Social Lab can be used by system administrators to access user-generated data in a graphical way—that is, without accessing the database engine directly. Social Lab allows CRUD (create, read, update, and delete) operations for 18 of the tables in Social Lab's data model. Thus, researchers can, for example, quickly retrieve the number of currently registered users, the number of public or private messages published (e.g., status updates or comments in users' walls), how many pictures were uploaded and which users were tagged in them, the number of fan pages created by users, and other kinds of summary information. The backend application also provides filters to refine the searches and to define more elaborate queries, such as the following: Which

Table 2 "Alice Johnson" bot script (first level of Social-Lab-based privacy wargame)

User	Step	Command	Automsg
Alice Johnson	0	Accept Friendship	
Alice Johnson	1	System Message	Level 2

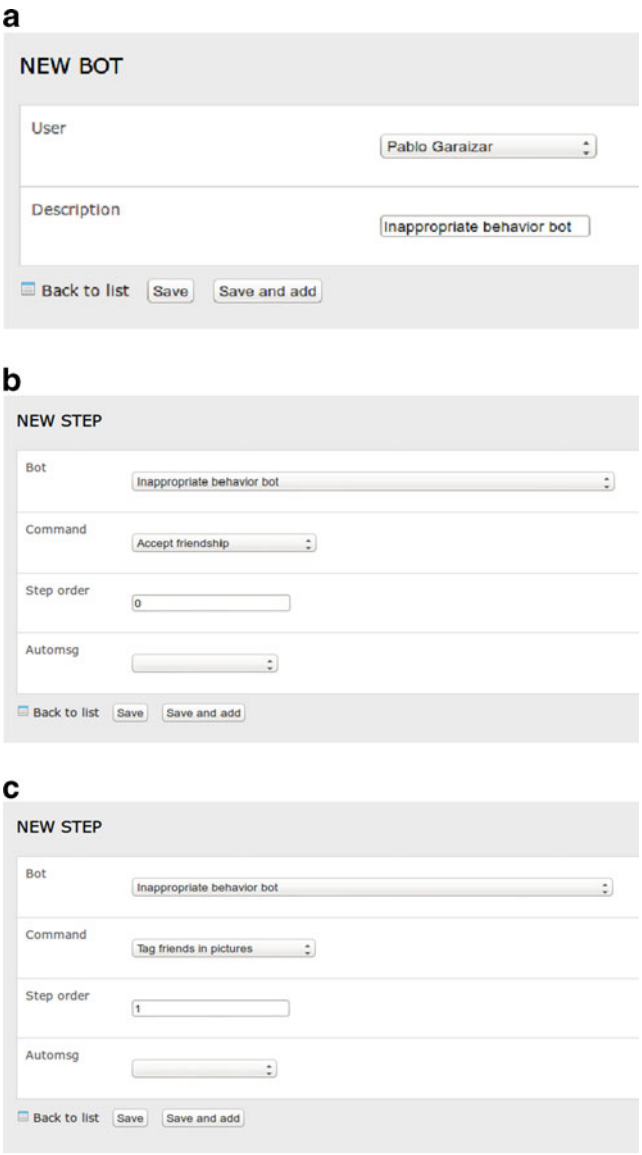

Fig. 3 Process of creating a new automated behavior. **a** Definition of the profile as a bot. **b** Definition of the first step of the procedure of the bot ("Accept friendship"). **c** Definition of the second step of the procedure of the bot ("Tag friends in pictures")

users have asked to become a friend of a specific user? (see Fig. 5). Furthermore, because this application allows access to the tables that store the behaviors of social bots, it is possible to use it to create new user tasks or modify the behavior of existing ones, as will be explained in the next section.

Third, for cases involving queries on the data of Social Lab that are too complex to be addressed from the backend application, there is always the possibility of access to the database through SQL. The schema.yml file in Social Lab's source code stores a YAML (Yet Another Markup Language; see Ben-Kiki, Evans, & Ingerson, 2009) representation of the data model of the social network (i.e., all the properties and types of the entities, relationships, and auxiliary tables needed to store user-generated content: user profiles, messages, friendship requests, pictures, tags, fan pages, etc.). On the basis of this schema, researchers can define queries involving multiple tables with complex conditions, joins, groupings, or other advanced features provided by SQL.

Example: Social Lab as a privacy wargame

As a specific example for the use of Social Lab, we created a social engineering wargame available at http://demo.sociallab.es. Social engineering, in the context of computer security, is a set of techniques designed to manipulate people into performing actions or disclosing confidential information. Social psychology and social engineering are closely related, because the success of the latter depends on a thorough understanding of human social behavior (Cialdini, 2001). Hackers and crackers take advantage of social engineering techniques to gain access to technologically well-protected systems (i.e., those using firewalls, intrusion detection systems, or other perimeter security solutions). At a less sophisticated level, social engineering can be used by strangers or stalkers to gain access to private information of a victim—for example, to deceive others (Whitty et al. 2012). Furthermore, a wargame, in this context, is a security challenge in which players must use their skills to exploit vulnerabilities in a system to gain access to it. Wargames often provide a set of levels of increasing difficulty to facilitate the learning process about defense against hacking (i.e., Hackerslab,[5] Try2Hack,[6] Hack this site,[7] etc.). Because it is privacy oriented, our example instantiation of Social Lab is a social engineering wargame.

Privacy in social networks

There are several reasons that have prompted us to address the problem of digital literacy regarding privacy in SNSs from a disruptive perspective. First, teaching online privacy is a difficult task as long as it goes in the opposite way to being social, and therefore, undermines the user experience in social media (Fagerlund-Savisaari, 2010). All measures aimed at preserving user privacy represent a usability loss in the SNS and are usually experienced by users as boring or annoying (Edbrooke & Ambrose, 2012). Second, privacy learning materials designed from a playful perspective are often intended to be used by children or teenagers (Johnson, 2011a, 2011b, 2011c). However, not only these age cohorts are lacking in knowledge on privacy; older users face similar problems, and they may feel uncomfortable learning with children-oriented materials (Cranor et al. 2007; Fischer-Hübner & Lindskog, 2001; Ovaska & Räihä, 2009). For these reasons, we decided

[5] http://www.hackerslab.org/eorg/
[6] http://www.try2hack.nl/
[7] http://www.hackthissite.org/

```
public function executeScheduler() {…
  foreach($tasks as $task) { …
    foreach ($steps as $step) { …
      switch ($step->getCommandId()) { …
        casesfConfig::get('app_command_checkfof'):
          if ($to->isFriendOfFriends($from)) {
            $task->setStep($task->getStep() + 1);
          } else {
            $donotcontinue = true;
          }
          break;
      …}
    …}
  …}
…}
```

Fig. 4 Excerpt of Social Lab's source code where command "Check friend of friend" is defined

to offer an interactive social game where players could learn social engineering techniques from a first-person perspective. Thus, as happens in training courses on ethical hacking, after playing with the privacy wargame, it may be easier for these users to identify situations where malicious others pretend to use the same techniques on them in real SNSs.

Motivating social interaction: From levels to game play

On the basis of previously described mechanisms, our privacy-focused instance of Social Lab provides a 10-level wargame of increasing difficulty. All the information of the profiles related to these 10 levels is fictitious, so no privacy of any real person is violated when playing Social Lab. Along these 10 levels, players will have to refine their social engineering techniques to solve the corresponding challenges. Players can collaborate with others to solve these levels—for example, share information about profiles or their own activity—or even facilitate the interaction between other players and social bots. In addition, players can compete against each other; strategies include avoiding communication about ways to solve levels, posting misleading information, or limiting access to information in one's profile. Players may also decide to cooperate with other players and work as a team. For research purposes, those players who agree to be part of research can be monitored for behaviors and strategies, they can be subjected to experimental manipulations (e.g., measures that support vs. suppress group decision making), and they can be interviewed with standard Web-based questionnaires (e.g., Reips & Birnbaum, 2011).

The Social Lab privacy wargame can be used individually via the Internet or within a training program or workshop about social networking where a teacher provides instructions or hints on how to advance in the game to complete it during the workshop (Nuñez, Garaizar, & Reips, 2013). Table 3 shows the set of social bots required to create the 10 levels of the example wargame.

Setup

Once the Social Lab source code is installed and configured (see Social Lab's home page for details), the researcher needs to prepare the content of the research scenario. In the case of the wargame on privacy, we created 30 user accounts with basic information filled in (i.e., profile details, comments on the wall, friend relationships among users, pictures, fan pages, etc.). The easiest way to accomplish this task is to enable the sign-up process and create accounts manually, using Social Lab's Web interface.

After the preparation of the user accounts, the researcher needs to access Social Lab's backend application to transform some of these user accounts into bots. Many behaviors (e.g., all those needed to be deployed in our example on privacy) are already defined as commands in Social Lab. The behavior of social bots can be configured by changing or adding rows in the Step table, as explained in the previous section. Finally, the researcher needs to configure a scheduled task at the server to launch the social bots manager periodically (i.e., via *cron* on UNIX-like systems or the *Task Scheduler* on Windows).

The state of all current settings of a research scenario can be exported to an external YAML file. Using this file, it is possible to create modified versions of research scenarios (e.g., a translated version in another language). These modified versions can then be reimported, and they can be exchanged with other researchers—for example, for the purpose of replication of research studies.

Extending Social Lab for other research purposes

Social Lab is designed to serve as a general purpose social media research application; with it, researchers can create specific instances of social networks for research purposes. As an example, we have presented an instantiation of Social

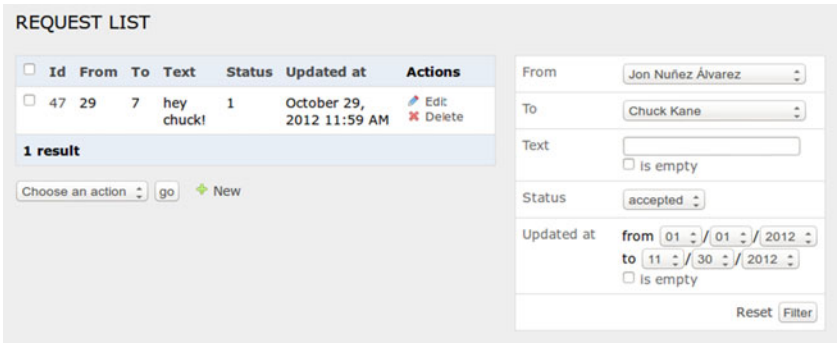

Fig. 5 Using the backend application's filters to query user-generated content in Social Lab

Lab as a platform for a social wargame for the purpose of learning about and investigation of privacy management in social networks.

Researchers may use Social Lab to facilitate social network analysis and to investigate group-level social network structure. Social network analysis is the methodical analysis of social networks—for example, research into changes of the pattern of connections between persons in a collective. It can be applied to social networks that are developed within Social Lab or manifest themselves in communications on this platform. With Social Lab, researchers can collect information on who is connected to and communicates with whom, in time. Such data can then be used to investigate social network properties, such as density, clustering, and connectedness, or node properties, such as betweenness or centrality (see D'Andrea, Ferri, & Grifoni, 2010, and Mislove, Marcon, Gummadi, Druschel, & Bhattacharjee, 2007, for further information on online social network analysis).

For example, the social network that develops within an instance of Social Lab can be observed in detail as it develops over time, including phenomena such as attitude polarization, group decision making, and minority influence. To study attitude polarization, for example, all postings of research participants within the social network can be tracked over time and be examined for polarization effects and also in response to information posted and to events inside or outside the network. Attitudes could also more explicitly be measured by a social bot that asks users for their opinions about an attitude object or even sends participants to a Web-based questionnaire. Similarly, this software can be used and extended to facilitate group processes—for example, for work teams within organizations. Experimental use of Social Lab could compare two instantiations that receive different interventions—for example, a social bot that routinely sends reassuring versus neutral messages following specific user actions.

Table 3 List of social bots needed to create the social engineering wargame at Social Lab

Id	Name	Description
2	Alice Johnson	Level 1: Accept always
3	Bob Smith	Level 2: Ask location
4	Carol Wang	Level 3: Ask request
5	David Danielson	Level 4: Friend of a friend
6	Elisabeth Benz	Level 5: Page match
7	Chuck Kane	Level 6: Academic match with limited profile view
8	George Godwin	Level 7: Friend tag match
9	Helen Hathaway	Level 8: Page owner match
10	Igor Magnuson	Level 9: Tag match
11	Jack Michaels	Level 10: Friend of two friends
12	Kate Johnson	Level 4: Always accept bot needed to have a friend of a friend
13	Laura Kurtz	Level 6: Always accept bot needed to gain access to profile limited to friends of friend
14	Linda Rommer	Level 9: Always accept and tag bot needed to persuade level9 bot
15	Mike Zimmerman	Level 10: Always accept bot to check wall
16	Nate Pierce	Level 10: Check wall, friend of level 10
19	Ryan Smith	Level 10: Friend tag match, friend of level 10

To ensure informed consent, Social Lab provides two features. First, in order to finish the process of creating a user account, a user has to accept the Terms of Use of the site. Researchers and research institutions can adapt the Terms of Use to their specific situation and needs. Furthermore, Social Lab, by default, offers an option in each user profile to opt out of participation in research projects. Requests to participate in research projects after filling in an informed consent form can be sent to users automatically from time to time. Second, researchers can create the required number of user accounts for their study and disable the sign-up process in Social Lab. Therefore, only those who obtained a user account directly from the researchers will be able to access the site.

Social Lab's source code is released under a free software license (Affero General Public License version 3). Therefore, researchers may freely download it, modify it, and share their modifications. Some of the images used in Social Lab are the property of David Niblack and are released under a Creative Commons Attribution 3.0 license. The development process of the platform is public and can be followed and contributed to by tracking the public code repository at GitHub.[8]

The open development model has favored the collaboration of freelance contributors, who review and patch the code. We published Social Lab in English, Spanish, and German—and within weeks, Juanan Pereira localized it to the Basque language. All the text strings used in the interface of Social Lab are stored in an external XLIFF (a XML dialect) file that supports ease of translation. Similarly, the descriptions and interactions of the fake users needed to create the privacy wargame and the system messages of Social Lab are stored in an external YAML file. Therefore, the whole platform can be translated easily by third-party contributors, who do not need to know anything else about its implementation.

Conclusions and outlook

In this article, we have presented Social Lab, a social network software designed for online research. The purpose of the example instantiation of Social Lab was to be a social engineering wargame focused on digital literacy in privacy. However, as we have seen, Social Lab can be programmed for other purposes and can be used as a laboratory for a wide range of studies on social networking—studies that, by nature, do not fall within the Terms of Service of general purpose social networks like Facebook, studies requiring more detailed and precise control of conditions, or studies that need to keep track of all interactions performed on the platform. For all these kinds of studies, Social Lab provides an open, adaptable, and continuously evolving solution.

We currently offer several services around Social Lab: (1) general information about the project at http://www.sociallab.es, (2) public access to the code repository at GitHub, and (3) demonstration servers in the officially supported languages (currently, English,[9] Spanish,[10] German,[11] and Basque[12]).

Looking ahead, our goal is twofold. First, we hope that more and more users will use our demo servers to assess the current instance of Social Lab about privacy. With increasing number of users of the platform, it will be possible to conduct large-scale studies—for example, multifactorial experimental designs with many levels. Second, we hope that other researchers can leverage our efforts in developing Social Lab to create their own online laboratories for social experimentation and overcome the limitations imposed by the companies behind general purpose social networks. Our aim is to provide a new tool for the scientific community to advance in the exciting study of behavior in social media.

Author Note We thank two anonymous reviewers and the editor for valuable feedback. Support for this research was provided by Cátedra Telefónica–Deusto. The authors would like to acknowledge the contribution of the EU COST Action IS1004 "Webdatanet" (http://webdatanet.eu). The authors declare that there was no conflict of interest in the publication of this study. Correspondence concerning this article should be addressed to Pablo Garaizar, Deusto Institute of Technology (DeustoTech) – Universidad de Deusto, Avda. Universidades 24, 48007, Bilbao, Spain. E-mail: garaizar@deusto.es

References

Acquisti, A., & Gross, R. (2006). Imagined communities: Awareness, information sharing, and privacy on the Facebook. In G. Danezis & P. Golle (Eds.), *Privacy Enhancing Technologies. Lecture Notes in Computer Science, Volume 4258, 2006, pp 36–58*. Heidelberg: Springer Berlin. doi:10.1007/11957454_3

Ben-Kiki, O., Evans, C., & Ingerson, B. (2009).Yaml Ain't Markup Language (YAML)(tm) version 1.2. YAML.org, Tech. Rep., September 2009.

Birnbaum, M. H. (2000). SurveyWiz and FactorWiz: JavaScript Web pages that make HTML forms for research on the Internet. *Behavior Research Methods, Instruments, & Computers, 32*(2), 339–346.

Buffardi, L. E., & Campbell, W. K. (2008). Narcissism and social networking Web sites. *Personality and Social Psychology Bulletin, 34*(10), 1303–1314.

Cialdini, R. B. (2001). *Influence: Science and practice (Vol. 4)*. Boston, MA: Allyn and Bacon.

Cranor, L., Hong, J., & Reiter, M. (2007). Teaching usable privacy and security: A guide for instructors. Retrieved May 1, 2013, from: http://cups.cs.cmu.edu/course-guide/

D'Andrea, A., Ferri, F., & Grifoni, P. (2010). An overview of methods for Virtual Social Network Analysis. In A. Abraham, A.-E. Hassanien, & V. Snasel (Eds.), *Computational Social Network Analysis: trends,*

[8] https://github.com/txipi/Social-Lab/

[9] http://en.sociallab.es
[10] http://es.sociallab.es
[11] http://de.sociallab.es
[12] http://eu.sociallab.es

tools and research advances (pp. 3–26). Berlin, Germany: Springer. doi:10.1007/978-1-84882-229-0

Edbrooke, O., & Ambrose, M. L. (2012). Teaching privacy in the twenty-first century. *Social Education, 76*(4), 217–220.

Fagerlund-Savisaari, A. (2010). Thanks for adding me! : The complexity of Facebook friendships and public privacy. Case: Finnish politicians. Tampereenammattikorkeakoulu. Retrieved May 1, 2013, from: http://publications.theseus.fi/bitstream/handle/10024/14558/Fagerlund-Savisaari_Anna.pdf

Fischer-Hübner, S. & Lindskog, H. (2001).Teaching privacy-enhancing technologies.In *Proceedings of the IFIP WG 11.8 2nd World Conference on Information Security Education*. Perth, Australia, pp. 1–17.

Göritz, A. S., & Birnbaum, M. H. (2005). Generic HTML Form Processor: A versatile PHP Script to save Web-collected data into a MySQL database. *Behavior Research Methods, 37*(4), 703–710.

Hauben, M., Hauben, R., & Truscott, T. (1997). Netizens: On the history and impact of Usenet and the Internet. Wiley-IEEE Computer Society Pr; 1 edition (April 27, 1997).

Humphreys, L. M., Krishnamurthy, B., & Gill, P. (2010). How much is too much? Privacy issues on Twitter. Paper presented at the 60th Annual Conference ofthe International Communication Association. Singapore.

Johnson, M. (2011a). From passport to MyWorld: Media awareness network extends digital literacy skills to secondary students. MediaSmarts, Media Awareness Network. Retrieved May 1, 2013, from: http://mediasmarts.ca/blog/passport-myworld-media-awareness-network-extends-digital-literacy-skills-secondary-students

Johnson, M. (2011b). Privacy pirates: An interactive unit on online privacy. MediaSmarts, Media Awareness Network. Retrieved May 1, 2013, from: http://mediasmarts.ca/blog/privacy-pirates-interactive-unit-online-privacy

Johnson, M. (2011c). Winning the Cyber security game. MediaSmarts, Media Awareness Network. Retrieved May 1, 2013, from: http://cira.ca/assets/Documents/Publications/WinningCyberSecurityGameLesson.pdf

Meeker, M., Devitt, S. & Wu, L. (2010, June 7), Internet trends, Morgan Stanley Research. Retrieved May 1, 2013, from: http://www.slideshare.net/CMSummit/ms-internet-trends060710final

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). Measurement and analysis of online social networks. Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, IMC '07, 29–42, New York, NY, USA.

Nuñez, J., Garaizar, P. & Reips, U.-D. (2013). Online workshop on privacy using Social Lab, a social engineering wargame. 7th International Technology, Education and Development Conference, INTED 2013, 4th-6th March, Valencia (Spain).

Ovaska, S. & Räihä, K.-J. (2009). Teaching privacy with Ubicomp scenarios in HCI classes. Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group. OZCHI 2009, 411, pp. 105–112. ACM, New York. 2009.

Reips, U.-D. (2001). The Web Experimental Psychology Lab: Five years of data collection on the Internet. *Behavior Research Methods, Instruments, and Computers, 33,* 201–211.

Reips, U.-D., & Birnbaum, M. H. (2011). Behavioral research and data collection via the Internet. In K.-P. L. Vu & R. W. Proctor (Eds.), *The handbook of human factors in Web design* (2nd ed., pp. 563–585). Mahwah, New Jersey: Erlbaum.

Reips, U.-D., & Buffardi, L. (2012). Studying migrants with the help of the Internet: Methods from psychology. *Journal of Ethnic and Migration Studies, 38*(9), 1405–1424. doi:10.1080/1369183X.2012.698208

Reips, U.-D., & Funke, F. (2008). Interval level measurement with visual analogue scales in Internet-based research: VAS Generator. *Behavior Research Methods, 40,* 699–704.

Reips, U.-D., & Garaizar, P. (2011). Mining Twitter: Microblogging as a source for psychological wisdom of the crowds. *Behavior Research Methods, 43,* 635–642.

Reips, U.-D., & Krantz, J. H. (2010). Conducting true experiments on the Web. In S. Gosling & J. Johnson (Eds.), *Advanced methods for conducting online behavioral research* (pp. 193–216). Washington, DC: American Psychological Association.

Reips, U.-D., & Lengler, R. (2005). The Web Experiment List: A Web service for the recruitment of participants and archiving of Internet-based experiments. *Behavior Research Methods, 37,* 287–292

Reips, U.-D., & Neuhaus, C. (2002). WEXTOR: A Web-based tool for generating and visualizing experimental designs and procedures. *Behavior Research Methods, Instruments, and Computers, 34,* 234–240.

Reips, U.-D., & Stieger, S. (2004). Scientific LogAnalyzer: A Web-based tool for analyses of server log files in psychological research. *Behavior Research Methods, 36*(2), 304–311.

Stieger, S., & Reips, U.-D. (2008). Dynamic Interviewing Program (DIP): Automatic online interviews via the instant messenger ICQ. *CyberPsychology and Behavior, 11,* 201–207.

Stieger, S., & Reips, U.-D. (2010). What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior, 26,* 1488–1495.

Watters, A. (2011). How recent changes to Twitter's terms of service might hurt academic research. *Read Write*, March 3rd, 2011. Retrieved May 1, 2013, from: http://readwrite.com/2011/03/03/how_recent_changes_to_twitters_terms_of_service_mi

Whitty, M. T., Buchanan, T., Joinson, A. N., & Meredith, A. (2012). Not all lies are spontaneous: an examination of deception across different modes of communication. *Journal of the American Society for Information Science and Technology, 63*(1), 208–216.