Online Neighborhood Watch: The Impact of Social Network Advice on Software Security Decisions Surveillance de voisinage en ligne: l'impact de conseil de réseau social sur les décisions de sécurité de logiciel

Bruna Freitas, Ashraf Matrawy, *Senior Member, IEEE*, and Robert Biddle, *Member, IEEE*

***Abstract*— Malicious software (malware) is one significant threat to Internet security. Malware is designed to harm a computer or network, and can be installed on one's machine without their consent. Attacks are often done by deceiving people into downloading malicious software that is posing as useful software. We speculated that if people had advice from a trusted source, they would be inclined to use the advice, reducing their chances of putting their computers at security risk. We designed and developed a system, Online Neighborhood Watch (ONWatch), to provide social network advice to users considering downloading software, sometimes offering alternatives when software was not trustworthy. We ran an empirical study to compare the advice coming from a trusted person to the advice coming from other more general social networks. We compared five different sources of advice in total. We did not find much evidence that the advice had a different effect based on the advisor, but the study confirmed our hypothesis that presenting alternative software will improve security.**

***Résumé*— Les logiciels malveillants (malware) sont une menace importante pour la sécurité de l'Internet. Le malware est conçu pour nuire à un ordinateur ou un réseau, et peut être installé sur la machine de quelqu'un sans son consentement. Les attaques sont souvent faites pour tromper les gens à télécharger un logiciel malveillant qui est présenté comme un logiciel utile. Nous avons spéculé que si les gens avaient un conseil à partir d'une source de confiance, ils seraient enclins à utiliser the conseil, ce qui réduit leurs chances de mettre leurs ordinateurs à risque de sécurité. Nous avons conçu et développé un système, Surveillance de Voisinage en Ligne (OnWatch), pour donner conseil de réseau social pour les utilisateurs qui envisagent le téléchargement de logiciel, offrant parfois des solutions de rechange lorsque le logiciel n'était pas digne de confiance. Nous avons effectué une étude empirique pour comparer le conseil provenant d'une personne de confiance au conseil provenant d'autres réseaux sociaux plus généraux. Nous avons comparé cinq différentes sources de conseils au total. Nous n'avons pas trouvé beaucoup d'évidences que le conseil avait un effet différent sur la base du conseiller, mais l'étude a confirmé notre hypothèse que la présentation de logiciel alternatif améliorera la sécurité.**

***Index Terms*— Computer crime, counterfeiting, crowdsourcing, internet, social computing.**

I. Introduction

MALICIOUS software (malware) is a significant threat to Internet security. Malware is designed to compromise a computer or network, and can be installed on one's machine without their consent or knowledge. If malware is installed in the victim's computer, the attacker can perform malicious activities, including having access to the user's files. The strategy of attack is often through deceiving people into clicking on links or downloading malicious software that is posing as useful software. Identifying malicious software can be a daunting task for ordinary users.

People do not purposefully download malware to their devices, and so attackers must use techniques to mislead users into downloading it unknowingly. In this paper, we intend to help people making better software security decisions by supporting decision making about the security of software, using advice from trusted people in their social network. People need to download software to help them work or play, and so it is not reasonable to simply forbid all downloads.

Inspired by the Neighborhood Watch program [1] (which we discuss in Section III), we speculated that a few people taking care of the computer security of an online "neighborhood" can, overall, improve security. If other people had such "neighborly" advice easily available to them, perhaps they would be more inclined to use the advice and lower their chances of being deceived into downloading malware. We developed software, Online Neighborhood Watch (ONWatch), to explore

Manuscript received September 23, 2014; revised August 14, 2015; accepted September 6, 2016. Date of current version December 6, 2016. This work was supported in part by the NSERC Strategic Research Network on Internetworked Systems Security. The work of A. Matrawy and R. Biddle was supported by the NSERC Discovery Grants Award.

The authors are with Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: brunamachado@gmail.com; ashraf.matrawy@carleton.ca; robert.biddle@carleton.ca).

Associate Editor managing this paper's review: Vahid Garousi.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/CJECE.2016.2613961

this idea. We compare the advice coming from a trusted person with the advice coming from other social networks, or from general people on the Internet (whom the users did not necessarily know). We also compare it with the advice coming from authorities. We focus on the effect of the source of security advice on how people behave, and explore the idea that social networks might aid, rather than hinder, security. We did not conduct field work on the actual security advice that people receive via social networks, but this might be an important topic for future work.

The applicability of this approach to security will depend on the characteristics of the contextual environment. For example, in enterprise settings, issues would likely include responsibility and liability for both giving and taking advice. Full exploration of these issues is beyond the scope of this paper, but will be important for future work.

Section II presents background to this paper, including the literature on similar previous research. We then clarify our research question and propose a solution framework. In Section IV, we overview the system to test our proposed solution. Section V presents our study and its methodology. Section VI presents the detailed analysis, and Section VII presents a broader discussion of the results. We then draw conclusions and identify opportunities for future research.

II. Background

A. Decision Making

Online advice services may be new, but the principles of advice are well established. People routinely have to make decisions and they often rely on advice from those they trust. In order to model such decision-making structures, the topics regarding advice giving and advice taking have been widely researched. Important decisions are rarely made by one person alone [2], and even for less important decisions, people also look for advice. People look for advice on decisions, such as which movie to see, so they might read critics' review, or a new college graduate is likely to consult their parents and peers about which job offer to accept. In most of the research about decision making and advice taking, the structure considered is the judge–advisor system ("JAS") [3]. In this structure there are two important roles. *The judge:* this term refers to the decision maker, which is the person who receives the advice and makes the final judgment. *The advisor:* this term refers to the source of advice, information or recommendation to the judge.

Sniezek and Swol [3] studied the influence of trust, confidence, and expertise in JASs in two experiments with judge–advisor pairs. Their research examined trust as the judge's attitude of relying on and using advice provided by the advisor during the decision task and as an expectancy about one's partner's behavior. Confidence, in their research, is the strength with which a person believes that a specific statement or decision is the best possible, and they believe that high advisor confidence can act as a cue to expertise and, therefore, influence the judge to accept the advice. Their experiments confirmed the importance of trust in the relationship of the judge and advisor in the acceptance of advice. Additionally, it confirmed the importance of cues to expertise, like confidence, in building trust especially when other information about the person is not available.

B. Related Work

Our approach has similarities with "crowdsourcing," a term coined to describe work outsourced to an online crowd of contributors [4]. There has been a variety of work exploring the use of crowdsourcing in computer security. Dong and Camp [5] outlined a framework for a broad range of ways in which self-organized communities might collaborate on the peer production of security information. Moore and Clayton study how crowdsourcing can be used to assist in identifying phishing sites, which are used to trick users into revealing online credentials. Liu *et al.* [6] later describe methods that can be used to refine this approach by computational techniques to improve accuracy and reliability. Neuber software's "security task manager" uses crowdsourced information to assist users in understanding the nature of processes running in Microsoft Windows, to aid the detection of unknown malware and rootkits hidden from antivirus software [7]. An alternative line of research explores security behavior "crowdsensing." For example, Burguera *et al.* have developed "Crowdroid," which captures application activity from Android devices allowing centralized analysis for malware detection [8].

There are two sets of work that resemble ours in how they aim to assist in the detection of malicious websites or software.

First, Web of Trust (WOT) [9] is a system that employs the wisdom of crowds to improve Web security. The service was launched in 2006 and works by advising the users which websites they can trust based on other users' experiences. It operates on the principle that a collective decision by ordinary users, when harnessed wisely, can yield good outcomes.

It takes the form of a centralized database and a browser add-on, working with the majority of browsers (Internet Explorer, Firefox, Google Chrome, Safari, and Opera). The WOT collects users' experiences by having them rate websites regarding four categories: trustworthyness, vendor reliability, privacy, and child safety.

Second, Chia *et al.* [10] designed and implemented a prototype system on the Nokia N810 tablet to show risk signals from a personalized community when installing applications in the tablet and to deter unsafe actions by slowing the user down with habituation-breaking mechanisms, such as a user clicking-through warnings. In this paper, they identify that the two currently prevalent methods to identify inappropriate software are the certification of "good" software by platform vendors and flagging of "bad" software by antivirus vendors or other global entities, which are both centralized. They believe that those centralized means of signaling appropriateness are ineffective and can lead to habituation (user clicking-through warnings) or disputes (users discovering that certified software is inappropriate).

They found that information during installation, such as End-User License Agreement (EULA), privacy policy, and disclaimer notices, is mostly ignored. Additionally, they found that security vendors, experts, and friends are important sources for information on digital risks. Moreover, 65% of the subjects in their research regarded the first-hand experience

by friends and family members as important. In comparison, fewer subjects considered the experience from the members of an online community to be important, suggesting that users regard inputs from friends and family members to be more relevant than those from an online community. They also found that when users know about digital risks, they are motivated to inform friends or family rather than the online community and that, in general, users consider reviews from trusted sources to be helpful.

Our approach builds on these ideas, but specifically explores the effect of advice from a trusted friend.

III. Research Question

Our interest is whether social networks improve security, and specifically explores advice from a trusted friend. The threat model we focus on is malicious software (malware) downloads posing as useful software, where the danger is that it gives some control of a user's computer to the malware creator, enabling them to do whatever they want. Our immediate research question was whether, in a controlled situation, users would be especially influenced by advice from a trusted friend, compared with advice from other sources.

People need to download applications that will help them accomplish the tasks that they need the computer for. In order to be safe, people must be careful when they download those applications from the Internet. People do not purposefully download malware to their machines, but when trying to download software that they need, but they might be misled into downloading malicious software or just poorly designed software that will allow attackers to gain control of their machines. People could, for instance, just avoid downloading all software, and they would then be secure from those types of threats, or they could just not connect to the Internet or even not bother buying a computer. Those are not practical solutions. Therefore, there is a tradeoff between being secure from Internet security threats and productivity: being able to accomplish whichever task people intend to do with their computer. These issues will be familiar to users with technical expertise, but for many ordinary users they present an important conundrum with no way to go forward.

People seek advice when making decisions ranging from the mundane (e.g., reading reviews in the newspaper before deciding on which movie to see or restaurant to go), to important issues (e.g., consulting with family and friends about taking a new job). Even when people have access to plentiful information, they often lack the ability to make sense of it, and they rely on the advice of trusted friends, colleagues, or advisors. We believe that most people are not computer security experts and will have a trusted person (a friend, a family member, and so on), who helps them when they have problems or questions regarding their computers. Hence, we suggest that one way in which social networks could improve security is that people could get advice from those trusted friends in order to make better decisions. We define *better security decision* to mean decreasing the number of malware downloads while still allowing downloads of legitimate software.

Our model is inspired by the Neighborhood Watch program. We believe that a few people taking care of the computer security of an online "neighborhood" can, overall, improve security. More details about our framework design will be described in Section IV. Another important factor in our design is that we believe that people have trusted friends, family members, or trusted acquaintances for specific issues. We suggest that people will have trusted connections for computer-related problems.

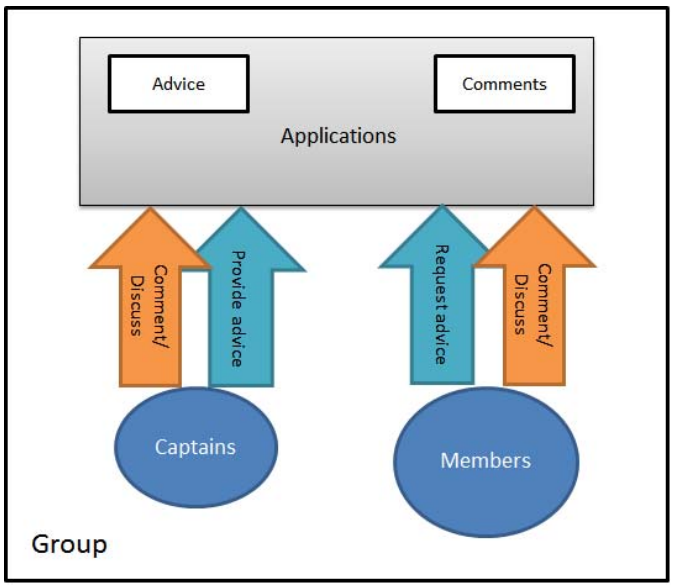

Fig. 1. ONWatch framework diagram.

Fig. 1 shows a diagram of our framework. The main focus of this framework is the advice. The objective of the framework is to facilitate the exchange of advice between the advisor and the person that is seeking advice.

When people are about to download software, they are usually on the download Web page, which usually contains information about that program, including its name. That is the moment in which our framework would provide the user with the advice. The user has had a chance to explore the Web site, and read the information they want when they get to this point. So, the advice should be presented on the download page, before the user makes the decision to download.

Our idea is that only people in a trust relationship join these groups, and therefore, we do not protect against malicious advice being propagated within the group. We also trust that people in the group make their best effort to only give advice if they feel that the advice is good. The issue of having bad advice being propagated in a group was out of the scope of this project.

In the real Neighborhood Watch program, there are two important roles: the watch members and the block captain [1]. The watch members are the backbone of the Neighborhood Watch program. Their duty is to look after their own best interests and those of their neighbors. They also must remain alert to the occurrence of any suspicious activity and report it promptly to the police, then to their coordinator. The block captain is principally responsible for monitoring a set of approximately ten homes within a Watch and to inform the residents of breakings and enterings or other threatening activities that have occurred in the area. In our framework, we

Fig. 2. ONWatch extension: the icons show the source, the advice, and a brief explanation. Here an advisor named Alex McSmart suggests the software Concertbox is a poor choice, with 70% confidence, and offers an alternative called GigFinder.

designate as "captain" the member responsible for providing advice. They are people who are interested in technology, and they have knowledge or interest in researching about malicious applications. We also believe that people will volunteer to be captain, because research shows that people make considerable sacrifices for the benefit of their group [13] [14].

Of course, captains might not have expertise in all software of interest to their group. We suggest however, that they will be able to leverage other sources of expertise: this is how social networks work. Moreover, our framework might be extended, as with real Neighborhood Watches, so that captains collaborate with other captains. It might seem that this network approach means users could do the same, but we believe that many users without any technical expertise are unsure where to look, or what to believe. We observe that for many, the first step is asking a trusted friend who may know, and if not will likely know where to look or who else to contact. This is the process we feel might be supported: the first trusted step. We acknowledge that people may not always wait to ask a trusted friend, and our approach does not address this. However, our observations are that people often do ask, and are at least sometimes willing to wait. In such cases, our approach will be beneficial, and thus can lead to harm reduction, even if there is no elimination of the problem.

The responsibility of captains should be taken seriously, as others in the group will expect trustworthy advice. In actual practice, the issues of liability and redress might even arise: such issues were beyond our research scope.

All this is motivation: the theory suggests such an approach might work, and our design would support the approach. Our next step was to conduct a study to explore this with users in a controlled setting.

A. Implementation

The implementation of our framework, ONWatch takes the form of a Web site and an extension for the Google Chrome browser. The Web site is used to manage information about the group itself, users, and captains, and to manage a database of advice about software.

In order for a user be able to see the advice without leaving the download Web page of an application, we have developed a Chrome browser extension, called the ONWatch extension. This extension is capable of retrieving the advice associated with the program that the user is considering and displaying it to the user as part of the body of the Web site (Fig. 2).

To use the extension, the user must be on the software download page, and then, highlight the name of the software they want to download and click the ONWatch extension. Once they do that, if the user is not logged in, a message displayed on the screen beside the highlighted name tells them to log in in order for them to see the advice. If they are logged in, the advice is displayed on the screen beside the highlighted name.

IV. Experiment

We designed and conducted an empircal experiment in order to investigate the research question: would users, in a controlled situation, be especially influenced by advice from a trusted friend, compared with advice from other sources? The independent variable was the source of advice to the participant, and there were five models: the "captain" model, a "friends from a social network" model, a "general people from the Internet" model, an "unpaid authority" model, and a "paid authority" model. Each of these models only differs in the source of the advice, and we explain each source in more detail below.

The experiment relied on role play, whereby we presented the participants with scenarios, discussed them with regard to the participant's own experience to contextualize them as best as possible, and asked them to act as they would be in that situation. This is not the ultimate way to explore our research question, but a truly ecologically valid approach would be extremely difficult to create in a controlled environment. We suggest that our approach is a reasonable first step necessary to inform further study.

Three hypotheses were made.

- H1: The captain model will lead to a better security decision than friends and general models.
- H2: The captain model will lead to similarly good security decisions as the authorities models.
- H3: Presenting alternatives will improve security.

Participants completed tasks related to deciding whether to download a certain application or not, and given advice. We measured: 1) the number of downloads; 2) the time taken to make a decision; 3) the compliance with the advice; and 4) the type of application being considered. Those are the dependent variables of the study.

Our study was approved by our University's Research Ethics Board. The participants in this study were recruited mainly around the university campus, via posters spread throughout the campus, as well as word-of-mouth. They were required to be over 18 years old and needed to feel comfortable with using a personal computer through a mouse and a graphical user interface. Using students as participants does limit the demographics, but their age and experience is very relevant for the risks involved in software downloads. They were compensated for their time in the form of a ten-dollar gift card from a popular coffee shop chain.

A pretest questionnaire provided us with information about the participants, including demographic information. We also presented the participants with a posttest questionnaire.

TABLE I
ADVICE

1.	Good Software	100%	no alternative
2.	Good Software	70%	no alternative
3.	Bad Software	100%	no alternative
4.	Bad Software	70%	no alternative
5.	No Information	not applicable	not applicable
6.	Good Software	70%	had alternative
7.	Bad Software	100%	had alternative
8.	Bad Software	70%	had alternative

To administer the experiment, the ONWatch Web site and Google Chrome extension were used. The Web site consisted of text that described the scenario that the participant would have to consider. It would provide the participant with background information regarding the motivation behind downloading an application or not. Following the scenario text, there would be the description of the software and an image that would relate to it. Finally, there would be two buttons to allow the participant to communicate whether they would download the referred application or if they would skip it. The Web site would present each participant with a total of 41 different scenarios with their respective application, being 33 belonging to the actual study, while the first eight were part of an initial practice set. The 33 programs used in the study consisted of 11 existing programs that were commercially available at the time of the study and 22 fictitious programs. The fictitious programs were created with names and functionality that would resemble other existing programs that were commercially available at the time of the study. The programs were categorized into three distinct types of software: well-known programs, utility programs, and entertainment (fun) programs. These groups were used for analysis, but not identified to the participant. Each program was associated with a set of attributes: a scenario, a name, a description, an image, and a related advice. We discussed these with the participants in advance to support the role-play aspects needed for the kind of software considered.

ONWatch (via the Google Chrome extension) provided the participant with security advice about the application in question. To test the effect of only the *source* of the advice, it was not based on any actual assessment nor related to the name or description of the software. Instead, the advice was set randomly as follows.

The advice consisted of a security assessment about the program (bad software or good software, each with one of two confidence levels, 100% or 70%, or no information), and a matching explanation. This means five types of advice. For the three levels, where advice was less than 100% good, we created an additional case, where alternative software was suggested. Whenever there was a piece of advice that would suggest an alternative, the advice for the alternative would always be "good software" with 100% confidence level. This resulted in eight levels of advice, as shown in the list in Table I.

Participants were randomly assigned to one of five conditions, where their advisor varied by condition. Those people who were in the captain condition were told to pretend that the advice was coming from their good friend who they trust for computer issues. The name of the advisor in this case was "Alex McSmart," because Alex is applicable to either male or female and also sounded intelligent. Therefore, it would make it easier for the participants to pretend that Alex was their good "tech savvy" friend. We again tried to support the participants in role play, suggesting that they think of one of their own trusted friends when considering the advice from "Alex."

Participants who were in the paid authority condition were told that they would receive advice from Symansoft, which is a (fictitious) security company and that the advice is a paid service, and they could assume that the fee had already been paid. Participants who were in the unpaid authority condition were told that they would receive advice from the (fictitious) Canadian Internet Security Agency (CISA), which was suggested to be a government organization that performs research in computer security and offers advice as a free service. For the social network condition, participants were asked to pretend that the advice came from their friends on a popular social network site. Finally, for participants in the "general people from the Internet" condition, they were told that they would receive advice from other users from around the world and they did not know them necessarily. The participants had to make $8 \times 3 = 24$ decisions in total, and all participants were presented with the same set of situations. A situation is defined as a pair of a scenario and application. Thus, the user had to read the scenario description and then read the program description. They had the option of reading the advice from their advisor and would finally make a decision to either download the program or skip it. As soon as they made their decision and selected the desired button, they would be presented with the next situation. Some pieces of advice, as mentioned before, could suggest to the participant an alternative program. In those cases, they would have to choose to download either one of the two options, or to skip both.

V. RESULTS: HYPOTHESES TESTING

The study took place over a one month period, with 1 h for each participant. More than 50 people were recruited, although we decided to use the data of 50 participants for the analysis, since some participants did not seem to fully understand the test. For instance, the participants received the following instruction: "please pretend those are real situations you are facing. Therefore, if the scenario says something, such as you really like puzzles, please pretend that you like puzzles even though in real life you might not." At the end of the study, a couple of participants seemed to have skipped most of the software in the fun category. When they were asked why, they responded that it was because they do not like games, or do not watch movies. So we inferred that they did not understand the test, and we did not use their data. This section reports the results. The dependent variables used in the analysis were number of downloads, compliance, and time taken to make a decision. All statistical analyses were carried out with the R language for statistical programming. Each of these hypotheses and the applicable results are described in the following. In summary, we found little support for hypothesis 1 and hypothesis 2 but strong support for hypothesis 3.

TABLE II
ALL DOWNLOADS

	Median	Mean	Min	Max	SD
Social Network	10.0	9.5	7	11	1.6
Alex	8.5	8.2	5	11	1.8
General	8.0	8.1	5	11	1.9

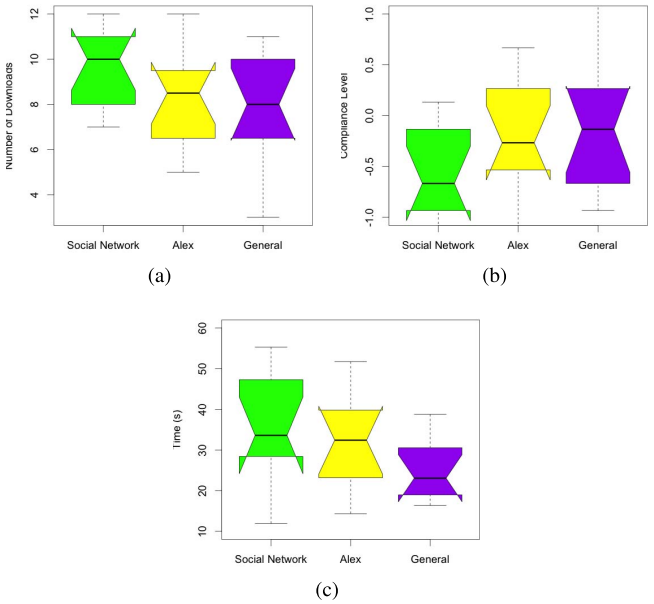

Fig. 3. Hypothesis 1 results. (a) All downloads. (b) Compliance level. (c) Average decision time.

A. Hypothesis 1

H1: The captain model will lead to better security decisions than the friends from social network model and the general people from the Internet model.

1) Downloads, Compliance, and Time: Table II shows descriptive statistics for the number of downloads. Fig. 3(a) shows box plots containing information about the total number of downloads. Box plots were chosen to illustrate the distribution of data because they can be a the convenient way of presenting data. They show the median as the black bar in the center, the second and third quartiles as the bottom and top edges of the box, and the first and fourth quartiles as whiskers extending from the box. The notches indicate 95% confidence intervals around the median. In the cases, where the confidence intervals fall outside of the second or third quartiles, the notches extend beyond the box. There can also be outliers, which are the points that appear to deviate markedly from other members of the sample in which it occurs.

No strong conclusion can be drawn just by looking in Fig. 3(a), but it is interesting to see that, overall, there are more people in Social Network choosing to download applications than people in Alex, while people in General present very similar behavior to people in Alex condition. Table II also shows that people in Alex and General have a very similar mean number of downloads, while for Social Network the mean is slightly higher. Also, it shows that everyone in Social Network choose to download at least seven applications, while people in the other two conditions had a lower minimum total number of download value of 5.

TABLE III
ANOVA OF ALL DOWNLOADS FOR SOCIAL NETWORK, GENERAL, AND ALEX

	df	SS	MS	F	p
Advisor	2	12.2	6.1	1.89	0.17
Residuals	27	87.0	3.2		

TABLE IV
COMPLIANCE

	Median	Mean	Min	Max	SD
Social Network	-0.7	-0.5	-0.9	0.1	0.4
Alex	-0.3	-0.2	-0.9	0.7	0.5
General	-0.1	-0.2	-0.9	0.7	0.5

TABLE V
DESCRIPTIVE STATISTICS FOR TIME (S)

	Median	Mean	Min	Max	SD
Social Network	33.6	35.4	11.9	55.3	13.3
Alex	32.4	32.6	14.3	51.8	11.9
General	23.1	24.6	16.4	38.8	7.4

We then conducted a one-way ANOVA test for difference between the three means. For this and other parametric tests reported in the following, we always confirmed distribution normality before proceeding. Table III shows the result of this ANOVA. No significant differences in total number of downloads between the three conditions were seen ($F(2, 27) = 1.89$ and $p = 0.17$)

Next the compliance level is analyzed for each condition. We tried to estimate how compliant each participant was with the advice by the following formula:

$$C(x) = \text{Adv}(x) - \text{Act}(x)$$

where

$$\text{Adv}(x) = \begin{cases} 2, & \text{if } x = \text{Y (Yes: good, 100\% confidence)} \\ 1, & \text{if } x = \text{MY (Maybe yes: 70\% confidence)} \\ 0, & \text{if } x = \text{NI (No information)} \\ -1, & \text{if } x = \text{MN (Maybe no: 70\% confidence)} \\ -2, & \text{if } x = \text{N (No: bad, 100\% confidence)} \end{cases}$$

and

$$\text{Act}(x) = \begin{cases} 2, & \text{if } x = \text{Download} \\ -2, & \text{if } x = \text{Skip.} \end{cases}$$

It is interesting to note that the more positive $C(x)$ is, the more cautious (more skips) the participant is. The more negative, the more risky (more downloads) the participant is.

Table IV shows descriptive statistics for the compliance. Fig. 3(b) shows box plots containing information about the compliance of members in each condition analyzed in this section. It shows that people in Social Network are the least compliant, and they tend to be more risky: they have clicked download more often. General and Alex are similarly compliant, although people in the General condition tended to be the most cautious. An ANOVA showed no significant difference between those three groups.

Finally, another interesting aspect to this analysis is the time to make a decision. We have calculated the time taken to

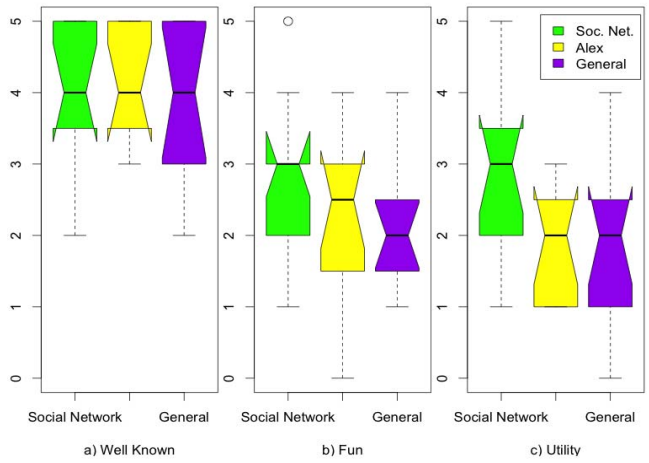

Fig. 4. Downloads in category.

make a decision for each scenario and the result can be seen in Fig. 3(c) and it is summarized in Table V.

We again conducted an ANOVA, which showed no significant difference in the time taken to make a decision between all the three conditions. Nevertheless, it is interesting to note that the time taken to make a decision by the participants in Social Network and Alex were both greater than the participants in General. For the latter, people in average were 10.8 s (30%) faster than in Social Network condition and 8 s (25%) faster than people in Alex condition. It is also interesting to note that in the General condition people were 10.8 s faster than Social Network, which is 50% of their average time. We would think that people make quick decisions when they know about something, therefore they do not need to spend time thinking about it. However, Fig. 3(b) shows that people in the General condition were actually very compliant with the advice. So, we speculate that they would check the advice and immediately follow that advice. That is a very surprising result, since our hypothesis was that people would trust a specific friend (Alex) better than general people from the Internet. We hypothesized that people in the Alex condition would make better software decisions than people in Social Network and General conditions. The results showed that, although people were slightly less compliant in the Social Network condition, no evidence of differences between those three groups was found. Therefore, we found little or no support for hypothesis 1.

2) Software Category: We also evaluated if there would be a difference in the number of downloads depending on the category of the software. As we talked previously, we came up with different applications that would fit in one of the three categories (FUN, UTILITY, and WELL KNOWN).

Fig. 4 shows the number of downloads per category of software. People in all the three conditions have downloaded more software in the WELL-KNOWN category than any of the other two categories.

Fig. 4 shows that in this experiment, people in the Social Network and General conditions have very similar behavior, while people in the Alex condition have a slightly higher download number in the WELL-KNOWN category. This suggests that people do not rely on advice or the tool (we observed

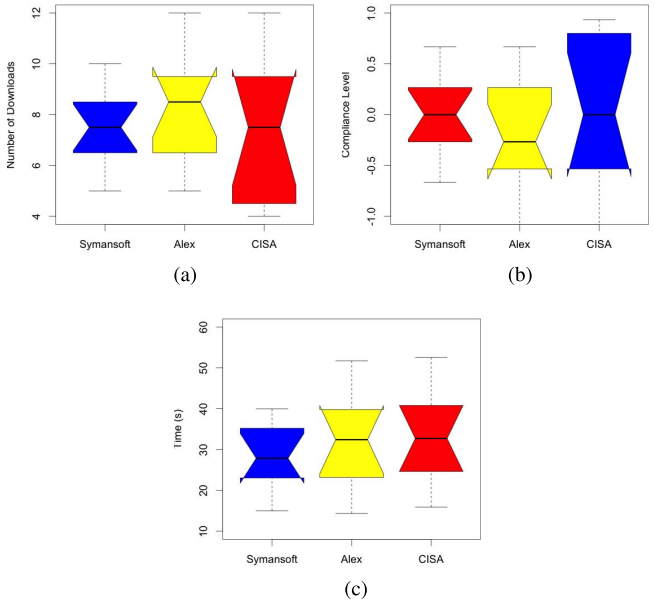

Fig. 5. Hypothesis 2 results. (a) All downloads. (b) Compliance level. (c) Average decision time.

TABLE VI
ALL DOWNLOADS

	Median	Mean	Min	Max	SD
Symansoft	7.0	7.1	5	9	1.4
Alex	8.5	8.2	5	11	1.8
CISA	7.5	7.6	4	12	3.0

during the laboratory sessions that sometimes participants would not even look at the advice) if they previously know the software. It is worth noting that the maximum number of downloads that each participant can have is five, meaning that in all the three cases, 25% of the people chose to download *all* the well-known software presented to them.

Research on decision making shows that decision makers often do not accept advice even if it might be beneficial, when they feel they can rely on their own judgment instead [15]. This might explain our results, because people tended to ignore the advice when they faced a *familiar* situation and trusted their own judgment whether it was safe or not to choose to download the well-known software.

B. Hypothesis 2

H2: The captain model will lead to similarly good security decisions as the paid security authority model and the unpaid security authority model.

1) Downloads, Compliance, and Time: Fig. 5(a) shows the total downloads chosen by each condition in this hypothesis, while Table VI shows the descriptive statistics for that data.

The mean number of downloads ranged between 7.1 in the Symansoft condition, and 8.2 in the Alex condition. The median number of downloads ranged between 7 in the Symansoft condition, and 8.5 in the Alex condition. The total number of available applications that people could choose to download was 15. Therefore, this result shows that, on average, participants chose to download half of the options available to them. The greatest number of downloads was 12,

TABLE VII
COMPLIANCE

	Median	Mean	Min	Max	SD
Symansoft	0.1	0.1	-0.4	0.7	0.4
Alex	-0.3	-0.2	-0.9	0.7	0.5
CISA	0	-0.03	-1.2	0.9	0.8

TABLE VIII
DESCRIPTIVE STATISTICS FOR TIME (S)

	Median	Mean	Min	Max	SD
Symansoft	27.9	28.5	15.0	40.0	8.5
Alex	32.4	32.6	14.3	51.8	11.9
CISA	32.7	33.3	15.9	52.6	11.9

and it was performed by someone in the CISA condition. Moreover, Fig. 5(a) shows that people in Symansoft were the most cautious. The maximum number of downloads in that condition was nine, which is the number of donwloads performed by 75% of people in both the Alex and CISA conditions.

A one-way ANOVA was conducted to test the equality of the three means. No significant differences in the number of downloads among the three conditions were seen, indicating that there was no evidence that participants behaved differently depending on their source of advice.

Table VII shows descriptive statistics for the compliance. Fig. 3(b) shows boxplots for the compliance level in each condition. It shows that people in the Symansoft condition were the most cautious, while people in the Alex condition were the most risky compared with the other two. An ANOVA was conducted with those three samples, but the results showed no significant differences.

Finally, Fig. 5(c) shows the boxplots of time in seconds taken by the participants to make a decision. The graphs for the Alex and CISA conditions look very similar, while people in the Symansoft condition made decisions slightly faster.

Table VIII shows the descriptive statistics of the time taken to make each decision for each condition. Mean times were similar and longest in the Alex (32.6 s) and CISA (33.3 s) conditions, and shortest in the Symansoft (28.5 s). The median times were shorter in every condition. A one-way ANOVA was conducted to test if there was a significant difference among the three groups. No significant differences in time were found among the three study conditions. All the tests made in this section showed that there is no differences among the Alex, Symansoft, and CISA conditions, which leads us to suspect that hypothesis 2 is supported: the Alex condition will lead to similarly good security decisions as the other two authority conditions. This result, however, is not definitive. Since the number of participants in each condition was not really high, we cannot confirm the results. More participants would be necessary for a more definitive result.

2) Software Category: In this section, we explore the influence of the software category in the number of downloads in each condition. Fig. 6 and Table IX show the number of downloads per category of software.

As expected, people in all the conditions have downloaded more software in the WELL-KNOWN category compared with the other two categories.

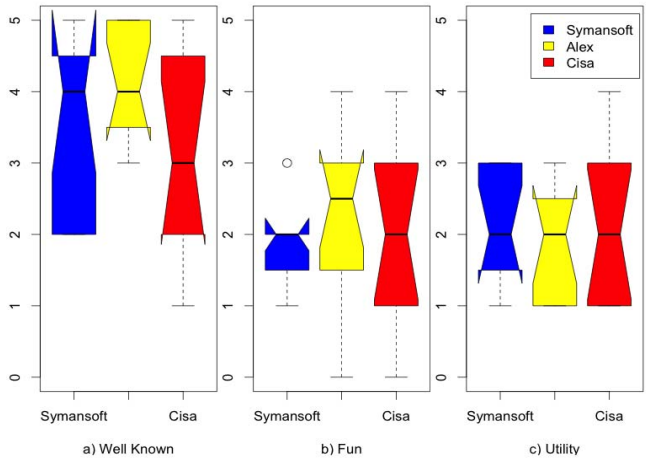

Fig. 6. Downloads in category.

TABLE IX
DESCRIPTIVE STATISTICS FOR NUMBER OF DOWNLOADS OF SOFTWARE IN EACH CATEGORY

WELL KNOWN					
	Median	Mean	Min	Max	SD
Symansoft	3.5	3.3	2	5	1.3
Alex	4.0	4.1	3	5	0.7
CISA	3.0	3.3	2	5	1.2
FUN					
	Median	Mean	Min	Max	SD
Symansoft	2.0	1.7	1	2	0.5
Alex	2.5	2.2	0	3	1.0
CISA	2.0	2.0	0	4	1.2
UTILITY					
	Median	Mean	Min	Max	SD
Symansoft	2.0	2.1	1	3	0.9
Alex	2.0	1.9	1	3	0.7
CISA	2.0	2.3	1	4	1.2

The median number of downloads for the WELL-KNOWN category was 3.0, 3.5, and 4.5 (Symansoft, Alex, and CISA conditions, respectively). For the other two categories, the median number of downloads was 2.0 (except for the Alex condition in the FUN category, which was 2.5). The maximum number of downloads in the WELL-KNOWN category was five for all the three conditions, while for the other two, it ranged from two to four. This clearly shows that people rarely trusted or used the advice for the software in the WELL-KNOWN category. Moreover, the minimum number of downloads in the FUN category was zero for the Alex and CISA conditions and one for the Symansoft. In the UTILITY category, the minimum number of downloads was one.

A two-way ANOVA was conducted in order to determine if there was any significant difference in the means of the independent variable analyzed in this section.

It showed that no significant differences in the number of downloads ($F(2, 81) = 1, p = 0.4$) were found among the Symansoft, Alex, and CISA conditions, even after applying any correction for posthoc analysis. Furthermore, the test showed that there is no significant difference in the interaction advisor *category ($F(4, 81) = 1, p = 0.4$). However, a significant difference was found among the categories ($F(2, 81) = 24, p < .001$). A pairwise t-test was performed in

TABLE X

t-TESTS OF SOFTWARE CATEGORY

	t	df	*p*
Well Known vs. Fun	6	57.0	<.001
Well Known vs. Utility	5.6	56.2	<.001
Utility vs. Fun	-0.55	57.9	0.586

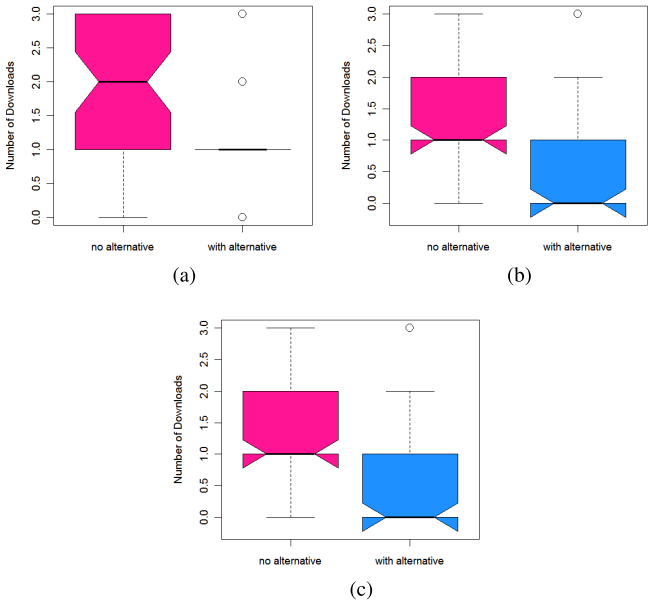

Fig. 7. Hypothesis 3 results. (a) MY Advice. (b) MN Advice. (c) N Advice.

order to determine which category is different from the others. The results can be seen in Table X. No difference was found between the UTILITY and FUN categories, but a significant difference was found between the WELL-KNOWN category and the other two categories.

C. Hypothesis 3

In this section, our last hypothesis and the data related to it are discussed. As previously cited, the hypothesis 3 is presented as follows.

H3: Presenting alternatives will improve security.

In order to be able to see if our hypothesis was supported or not, every participant had nine situations in which they were given a second option (alternate software) that they could choose from, in addition to the nine presented without alternative. Those situations happened when the advice was either MY, MN, or N. Whenever an alternative was suggested, this alternative would come with an advice of type Y. These same types of advice would also show up to the participant but with no suggested alternative. Thus, we were able to compare the number of downloads for each type of advice when there was a suggested alternative or not.

Fig. 7(a)–(c) shows boxplots for each advice given, and compares the cases with or without an alternative application. Except for Fig. 7(c), each pair of boxes shows differences, and the number of downloads decreases when the advisor suggested an alternative. Moreover, the confidence intervals of each box do not overlap with the other, suggesting distinct confidence intervals.

TABLE XI

DESCRIPTIVE STATISTICS FOR NUMBER OF DOWNLOADS OF SOFTWARE

Advice		Mdn	Mean	Min	Max	SD
MY	No Alt	2	2	0	3	0.9
	With Alt	1	1	0	3	0.8
MN	No Alt	1	1	0	3	0.9
	With Alt	0	0.6	0	3	0.8
N	No Alt	1	0.6	0	3	0.7
	With Alt	1	0.8	0	3	0.7

TABLE XII

t-TESTS OF ALTERNATIVE VERSUS NO ALTERNATIVE FOR ADVICE MY

Advice		*t*	df	*p*
MY	Alt vs. No Alt	5.5	97.2	<.001
MN	Alt vs. No Alt	3.6	97.5	<.001
N	Alt vs. No Alt	-1	97.0	0.314

Table XI shows the descriptive statistics for the number of downloads for each type of advice, both for when there was and when there was not an alternative. It is interesting to note that, for every type of advice, with an alternative or not, there were people choosing to download all the three possible applications and people choosing to skip all of them (min = 0 and max = 3 in all the cases). The median value was always lower when there was an alternative, except when the advice was N. In that case, the median number of downloads was 1 for both. Moreover, 75% of the participants downloaded 1 or less. Additionally, the fact that the median value was 1 or higher for all the cases, except when the advice was MN with an alternative, is probably explained by the downloads in the WELL-KNOWN category. As discussed in Section II, people tended to disregard advice when they were familiar with the software and download it. A t-test was conducted to determine if there was any significant difference in the means of the sample analyzed in this section. Table XII shows a summary of the results of these tests. A significant difference was found for the advice MY, when compared with and without an alternative ($t(97.2) = 5.5, p < .001$). A significant difference was also found for the MN advice ($t(97.5) = 3.6, p < .001$). Finally, no significant difference was found when the advice was N ($t(97) = -1, p = 0.314$).

From the obtained results, presenting an alternative to the user clearly improves security, assuming there is strong confidence in the suggested alternative. The only case where no difference was found when an alternative was suggested was when the advice was N. Many participants did choose to download an application when the advice was N, but most of those downloads were software that belonged to the WELL-KNOWN category. The beneficial effect of suggesting alternatives may seem obvious, but in online rating systems, it is often ignored. Our results show that including it might have a very positive effect. They also suggest that where the software may seem well known there may be challenges in making sure warnings are recognized.

VI. RESULTS: PERCEPTION

The participants' age in this study ranged from 18 to 43 years old. The median was 21 years old and the mean was 22.9. There were 26 male participants and

24 females, and all of them were students. Moreover, 44 out of the 50 participants were undergraduate students. There was a question asking if they had had some type of security training, and only three of all the participants answered that they had. One of them answered that his major in school was in network security and he is also working as a network technician. The other two mentioned that they had some short security training at work. They were also asked how often they browsed the Web, and 47 participants answered "daily," while three answered "several times a week." An interesting result came from the question "have you ever downloaded bad software to your computer?" Twenty-three participants answered yes. It is interesting that some of them describe their experience as their computer becoming slow after they downloaded the malware. Also, most of them described that they got viruses when downloading torrents, shareware, and kids' games.

The majority of the participants use their computer for entertainment as well as work. Less than 20 participants showed concern about having their computer compromised (answered one or two on a 10-point scale), while seven participants showed no concern (answered nine or ten). Participants also generally believe that an antivirus program protects their computer from any malicious application, and most of them believe that they know about the threats involved when downloading software to their computer. A surprising result came from the statement "I have a trusted person, who always helps me with computer issues." The histogram of responses was uniform, without any commonality in the answer. Overall, this study supposed that people had a trusted person that they would ask for advice about computer security and other issues. The participants' answers tell it differently.

In the posttest questionnaire, participants were asked a series of questions about their perceptions of usability, their trust in the advice, and in the system. The great majority of the questionnaire was made of statements, where the participants had to rank their agreement on a Likert scale from 1 to 10, like in the pretest questionnaire. The results showed that amongst the different conditions, there was broad agreement in the fact that they thought it was easy to use the tool, with some potential exceptions in the Alex condition. We speculate that, since they were told to pretend it was their trustworthy friend that was giving them the advice, they found it too difficult to imagine a friend whom they did not have.

One very interesting result from the study was that advice is not very effective when it is related to well-known software. Some comments participants made with regards to this issue were: "I know and use programs like Excel and would trust them to be safe despite the advice" and "because of my experience I knew some of the software and companies, so no need for advice." These results show that people have a poor understanding of the dangers of software that may only appear to be well known, and will override even the advice of a trusted friend. This suggests that for well-known software, it is especially important for negative advice to be specific: for example explicitly suggesting the software may have been subject to tampering.

VII. Conclusion

To the best of our knowledge, this is the first study that investigates the relationship between advice from a trusted friend and malware downloads. It presents an early step in the area of security advice in online social networks. We proposed, implemented, and tested a framework to support decision making about the security of software, utilizing advice from trusted people in their social network, and to understand how our current social networks can improve computer security. Our empirical study compared the effect of advice coming from a trusted person with the advice coming from other more general social networks, and the different advisors were the independent variables of the study. There were five in total: Alex McSmart (the trusted expert friend), friends from a Social Network, general people from the Internet, CISA, and Symansoft. We did not find much evidence that the advice had a different effect based on the advisor, but the study confirmed our hypothesis that presenting alternative software will improve security. This means that when people are told not to download a piece of software, they will likely avoid that download if they are presented with an alternative program that has similar functionality but better security. Advice was not very effective when it was about well-known software; such advice may need to be very specific about the potential threats involved.

References

[1] Ottawa Police Service. (Apr. 2003). *Neighbourhood Watch, Member Guide*. [Online]. Available: www.ottawapolice.ca,

[2] S. Bonaccio and R. S. Dalal, "Advice taking and decision-making: An integrative literature review, and implications for the organizational sciences," *Org. Behavior Human Decision Process.*, vol. 101, no. 2, pp. 127–151, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0749597806000719

[3] J. A. Sniezek and L. M. V. Swol, "Trust, confidence, and expertise in a judge-advisor system," *Org. Behavior Human Decision Process.*, vol. 84, no. 2, pp. 288–307, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0749597800929261

[4] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, Jun. 2006.

[5] Z. Dong and L. J. Camp, "Peersec: Towards peer production and crowdsourcing for enhanced security," in *Proc. Conf. Hot Topics Secur.*, HotSec'12. Berkeley, CA, USA, 2012, p. 8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2372387.2372395

[6] G. Liu, G. Xiang, B. A. Pendleton, J. I. Hong, and W. Liu, "Smartening the crowds: computational techniques for improving human verification to fight phishing scams," in *Proc. Symp. Usable Privacy Secur. (SOUPS)*, New York, NY, USA, 2011. Art. no. 8. [Online]. Available: http://doi.acm.org/10.1145/2078827.2078838

[7] A. M. Neuber. (2011). Security task manager. Software GmbH, Halle, Germany, [Online]. Available: http://www.neuber.com/taskmanager/

[8] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices (SPSM)*, New York, NY, USA, 2011, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/2046614.2046619

[9] Web of Trust. *Web of Trust*. (Sep. 2012). [Online]. Available. http://www.mywot.com/

[10] P. Chia, A. Heiner, and N. Asokan, "Use of ratings from personalized communities for trustworthy application installation," in *Information Security Technology for Applications* (Lecture Notes in Computer Science), vol. 7127, T. Aura, K. Järvinen, and K. Nyberg, Eds. Berlin, Germany: Springer, 2012, pp. 71–88.

[11] B. Arief, A. P. A. van Moorsel, D. Greathead, and L. M. Coventry, "Towards the implementation of an internet-based neighbourhood watch scheme-impacts of inclusive technologies on societies," in *Proc. CASON*, 2011, pp. 25–30.

[12] T. Bennett, K. Holloway, and D. Farrington, "The effectiveness of neighborhood watch," *Campbell Systematic Rev.*, 2008.
[13] P. Adams. (2012). Grouped: How small groups friends are key to influence social Web. New Riders, San Francisco, CA, USA. [Online]. Available: http://www.amazon.ca/Grouped-groups-friends-influence-social/dp/0321804112
[14] A. D. Shaw, J. J. Horton, and D. L. Chen, "Designing incentives for inexpert human raters," in *Proc. Comput. Supported Cooperat. Work (CSCW)*, New York, NY, USA, 2011, pp. 275–284. [Online]. Available: http://doi.acm.org/10.1145/1958824.1958865
[15] L. M. V. Swol, "Forecasting another's enjoyment versus giving the right answer: Trust, shared values, task effects, and confidence in improving the acceptance of advice," *Int. J. Forecast.*, vol. 27, no. 1, pp. 103–120, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169207010000415

Bruna Freitas received the B.Eng. degree in electronics engineering from the Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil, in 2007, and the M.A.Sc. degree in systems and computer engineering from Carleton University, Ottawa, ON, Canada, in 2012.

Since then she went on to gain experience and contribute to the private industry, while maintaining interest in the area of human factors in software design. She is currently a Software Engineer in Ottawa.

Ashraf Matrawy (S'98–A'02–M'03–SM'07) is currently an Associate Professor with the School of Information Technology, Carleton University, Ottawa, ON, Canada. He is also a Network Co-Investigator of the Smart Cybersecurity Network (SERENE-RISC), Montréal, QC, Canada. His current research interests include reliable and secure computer networking, software defined networking, and cloud computing.

Prof. Matrawy has served as a Technical Program Committee Member of IEEE Communications and Network Security (CNS), IEEE International Conference on Communications (ICC), IEEE Globecom, IEEE Local Computer Networks (LCN), and IEEE/ACM Cluster, Cloud and Grid Computing (CCGRID). He serves on the Editorial Board of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS.

Robert Biddle (M'02) received the B.Math. degree (Hons.) and the M.Math. degree from the University of Waterloo, Waterloo, ON, Canada, in 1977 and 1979, respectively, and the Ph.D. degree from the University of Canterbury, Christchurch, New Zealand, in 1987.

He is currently a Professor of Computer Science and Cognitive Science with Carleton University, Ottawa, ON, Canada. His current research interests include human factors in software design, especially issues relating to computer security.

Dr. Biddle is a Fellow of the New Zealand Computing Society. He is a member of the ACM.