Control and Understanding in Malicious and Benign Software

Eric Spero, Milica Stojmenović, Sonia Chiasson, Robert Biddle
Carleton University
Ottawa, Canada
eric.spero@carleton.ca

***Abstract*—This paper presents a study examining mental models of malware and regular software, in search of deep misunderstandings about malware and software which can be used in the design of new software and educational material. The study involved both a questionnaire, and two diagramming exercises. We decided to use a diagramming exercise because it is an effective medium for expressing spatial information which is important to mental models, and can get lost in verbal reports. Ours is the first study to examine mental models of malware using this technique. For the diagramming tasks, participants were asked to draw their understanding of how a word processor and malware work, respectively. Several key patterns emerged. General knowledge about malware, shown in the questionnaire responses was reasonable, but the deeper understanding of how malware functions, shown in the drawings, was concerning. Participants showed lesser knowledge of malware compared to regular software, and they seemed to regard malware as a fundamentally different kind of entity than regular software. They made black-and-white distinctions between malware and regular software in terms of whether the software is helpful or harmful, who the software serves, and who controls it. We discuss how these findings relate to decision-making online, and suggest that it might be beneficial to increase support for the control users have over their software. We speculate this might better equip users to make safe decisions surrounding software, thereby decreasing the effectiveness of malware.**

***Index Terms*—cybersecurity, malware, mental models**

I. Introduction

Using the internet exposes people to malware threats such as spyware, ransomware, and trojans. Users become affected by malware as a result of the decisions users make online: it is thought that unsafe decisions by end users are the cause of 80% of computer security problems [22]. Humans reason and make decisions using small-scale representations of the world called mental models [8], [15], [23]. When we are making the kinds of decisions that could lead to the installation of malware, our understanding of how computers and software work play an important role [5], [23], [31], [32].

With this study we aim to learn more about how people think about malware and related concepts, with the hopes of discovering something that would help us support users through new software design and educational material. We think that understanding what people think might help explain why they act certain ways. In addition, knowing what the gaps are in their understanding of malware could illustrate what type of support they would most readily benefit from, to help them make better and safer software-related decisions.

We elected to use diagramming exercises as the focal point of our data collection. Mental models frequently have a pictorial quality which are better expressed visually than with a verbal report [16]. Mental models capture the richness of experience in at least five dimensions: space, time, causality, entity, motivation (plans/goals) [36], and graphical depictions have proven to be effective for communicating this type of information. Diagramming tasks have been used in other studies of mental models of computer security (e.g. [17], [35]), but not yet for malware. We think the novelty of using diagramming exercises in this context, combined with its other useful properties will generate new insights into how people think about malware.

Mental models are not standalone entities, but rather are dependent on, and have implications for other concepts [37]. Since malware is a type of software, it would be useful to contrast how people think about malware with how they think about regular (or "safer") software. Prior research has found it useful to compare mental models with a reference model [25], but to our knowledge, this comparative method has not been attempted in the context of mental models of malware.

Our first research question is exploratory in nature—*RQ1: What are peoples' mental models of malware?* The present study is the first to elicit mental models using a diagramming activity, and the first to compare mental models of malware with mental models of regular software. We also probe for more information using a questionnaire to assess participant understandings of the purposes, effects, and methods of transmission of malware. Ultimately, we hope to be able to apply our findings to help users make safer decisions online. Our second research question is *RQ2: Are there characteristics of people's mental models which suggest how systems can be designed so that people are less susceptible to malware?*

II. Background

A. Mental Models

The notion that the mind represents and reasons about the world using internal models is usually traced back to Craik [8]. Craik argued that the mind "imitates" or "parallels" reality: we observe the external world and translate what we see to an internal symbolic representation with the same "relation-structure"—which means they both work in the same way. This functional equivalence between mind and model means that mental models can be used to simulate, and thus make

predictions about the real world. For example, using internal models we can build bridges in many different ways and perform stress tests on these designs to evaluate which is the best. This method is much quicker, cheaper, and safer than constructing many variations of actual bridges. The term *mental models* was popularized by Johnson-Laird [15], who has developed explicit, computable mental model–accounts for observations of how human reason in everyday situations. Norman [23] applied the idea of mental models to user interface design, and makes the point that mental models of a given piece of software can vary from user to user, and that what makes an adequate mental model is whether it supports successful interactions with the software.

Mental models are simplified representations, containing only the elements most relevant to a situation. What defines the adequacy of a mental model is whether it supports successful interactions with the external world [15], [23], and the amount of accuracy needed in a given mental model is therefore goal-dependent. To successfully use a television to watch a movie a model of "magic moving picture device" is satisfactory, but to repair a television one's model must include knowledge of its internal components and how they interact with each other.

Mental models simulate specific entities and events—tokens rather than types [3], and events are the "basic unit" [36]. Mental models are generated from deeper structures called *frames* [3], which are built out of perceptual experience. The richer our experience of an event or entity type, the greater our ability to generate useful mental models, and the more adaptive our behaviour.

In summary, mental models and the environment are inter-dependent. *Mental models shape our expectations of what will happen in the world, and what happens in the world in turn shapes our mental models*.

B. Mental Models in Computer Security

Mental models have been identified as a central issue in a number of studies in cybersecurity. For example, poor mental models of encryption have prevented users from adopting encryption tools [1], and weak mental models of wi-fi have given users a false sense of security, enabling risky behaviour [19]. There are large differences in the mental models related to computer security between security experts and non-experts [2], and users with lower computer literacy also have less sophisticated mental models related to computer security [4], [16]. Mental models are not fixed, but are constantly being updated [24]. A study on mental models of web certificates suggests that it is possible to help users make updates to their mental models that help them better understand computer security issues [27]. The work of Camp [5] emphasizes the metaphorical power of mental models, which allow users to understand things they do not know in terms of things they do, and she advocates for leveraging this property of mental models in communicating computer security risks to users.

The most notable work on the study of mental models of malware was conducted by Wash and Rader [31], [32]. Through a series of semi-structured interviews with typical computer users, Wash identified a number of "folk" mental models which people use to think about malware—particularly regarding the purposes, effects, and methods of transmission of malware [31]. These mental models also correlated with participant ratings of the importance of following security best practices.

C. Diagramming Exercise Rationale

Diagrams are useful both for facilitating and communicating thought. Using diagrams allow us to "off-load" information onto the environment, freeing up working memory to think about a problem more deeply than would be possible unaided [18]. Graphics have been used to communicate spatial information in the form of maps since ancient times, and more recently they have been used to communicate abstract ideas in a wide variety of contexts [29].

Earlier we noted that the basic unit of mental models is an event. Events vary along at least five dimensions: time, space, causation, entity, and motivation (goal/plan structure) [36]. A study examining mental models should use a format that allows users to easily communicate their understanding in terms of these dimensions, and we think diagrams provide this format.

Diagrams are elements systematically arranged in space [29], so they excel as a means of communicating spatial information. Spatial cognition is fundamental for our survival [12], and spatial structures play an important role even in non-spatial, abstract tasks [11]. For example, spatial metaphors underpin many common concepts (e.g. "I look forward to a brighter future") [21], and there is evidence that humans reason logically by creating a spatial array relating entities to each other [9]. This inherent advantage of spatial communication could allow for the easy expression of key information which might get lost otherwise.

Diagrams have long been used for communicating causal information informally [13]. Causal diagrams have also been shown to be effective in epidemiological research, and with "broadly intuitive appeal" [13]. Diagrams have been used to communicate complex entity-relations in a way that is easily understandable by people of a wide range of domain expertise [7]. Intelligence analysts use diagrams to capture entity-relationships, and to sequence events in time [30]. Maps are culturally universal tools [26] for mediating between the inner and outer world [14], and are some of the oldest forms of communication [14]. UML diagrams are used in business to represent relationships between agents and their goals [10]. Finally, Venn diagrams are often used to represent the logical relations between categories [34].

It is also possible to augment diagrams with verbal information, as we allowed our participants to do. Verbal-graphical representations like these allow for more communicative flexibility than solely written or drawn diagrams.

III. Method

Our research study included a demographics questionnaire, followed by two drawing tasks (described below), and finally a

post-task questionnaire on the purposes, effects, and methods of transmission of malware. This study was cleared by our Research Ethics Board (Clearance #109781).

Forty volunteers (18 females, 22 males) participated in the study. We used word of mouth for recruitment, with an effort to include a diverse community in terms of age, gender, occupation, and academic background. Eleven were aged 18-24, 12 were 25-30, 10 were 31-40, and the remaining seven were aged 41-70. Their educational backgrounds were mixed, with nine computer scientists and 31 of various non-computer science backgrounds (e.g. cognitive science, education, business, engineering, the humanities, and several with no academic background). Nineteen said that they had accidentally installed or had experienced problems due to malware at least once in their lives, 11 were uncertain, and 10 said that they had not. When asked if they knew someone (e.g. family or friend) that had accidentally installed or had problems with malware, 18 said "yes", eight "no", and 14 were uncertain.

A. Procedure

After completing the demographics questionnaire, participants received a prompt to draw a diagram (printed below). When they were satisfied with their drawing, participants were given a second drawing prompt (below). Thirty-nine participants used paper and pen for the drawings, and one used Microsoft Paint. Following the drawing tasks, participants completed a questionnaire featuring Likert-scale[1] questions on the purposes, effects, and methods of transmission of malware. Each participant completed the study separately, and it took approximately one hour each.

B. Drawing Prompts

We used two sets of drawing prompts—one for the first 20 participants, and the other for the second 20. The reason for this is described in the next subsection.

Prompt set 1:

- Word processor: *"Using a paper and pencil, please draw your understanding of how a word processor (e.g. Microsoft Word, Apple Pages, LibreOffice Writer) works to let users write and open text documents. In other words, try to draw what is happening inside the computer when you use a word processor to write and open text documents."*
- Malware: *"Say you install a word processor that contains malware. Using a paper and pencil, please draw your understanding of what you would expect this malware to do inside your computer."*

Prompt set 2:

- Word processor: *"Please draw a diagram to explain how a word processor (e.g. Microsoft Word) works."*
- Malware: *"A word processor has been affected by malware. Please draw a diagram to explain how this software works."*

After each prompt, we included the following disclaimer: *"The purpose of this exercise is to help us understand what you have in mind. This is not a test of artistic ability, and there are no 'correct' ways of drawing this picture."*

Our diagramming exercise asked participants draw only one type of malware: a Trojan. We chose a Trojan for the target malware type because we assumed most users would be familiar with it, and we wanted to avoid situations where the participant was unable to draw anything whatsoever.

C. Drawing Prompt Reformulation

After 20 participants completed the study, we were concerned about the possibility that the wording of our drawing task prompts could be biasing participants' drawings. The first potential issue with these prompts was that the word processor prompt was more specific than the malware prompt, which we thought might lead to more specific word processor drawings. The second potential issue was that the malware was described as something that is "contain[ed]" by the host software. This may have suggested a particular conception of malware.

We formulated the second set of prompts to address these two potential issues. After another 20 participants completed the study using the new prompts, we carefully inspected the two groups of pictures looking for major differences which could be attributed to the prompts, but did not find any. We therefore grouped all participants together in the final analysis.

D. Analysis Plan

We perform an exploratory analysis of the Likert-scale questionnaire results. Because one of our goals is to help improve user mental models of malware, we look for patterns that might reveal gaps in participant knowledge of malware.

We analyzed the drawings created by participants using grounded theory [6], [28], a bottom-up methodology for qualitative data analysis. In the grounded theory approach, analyzing the data occurs in stages, which are called open coding, axial coding, and selective coding. Open coding involves semantically tagging the data at the lowest level of meaning that the researcher is interested in. Codes are grouped into higher-order units called categories. Axial coding involves making connections across categories. Selective coding isolates a few codes as particularly important; it is the "driving force" behind the story the data tells. The process of selective coding culminates in a theory which provides a deeper explanation for the data.

IV. Results

A. Questionnaire

The boxplots in Figs. 1 to 4 show a summary of the results of the post-task questionnaire. Each of the questions shown here were answered using a Likert-scale response, where 1 stands for "very unlikely" or "strongly disagree", 5 for "very

[1]Likert scales are widely used in cognitive science research, and are thought to be good tools for quantifying information about individuals' attitudes [20], [33].

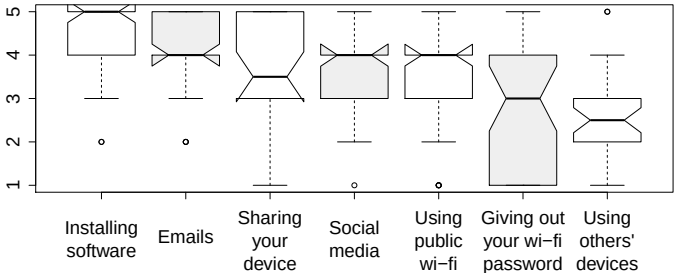


Fig. 1. Where does malware come from?

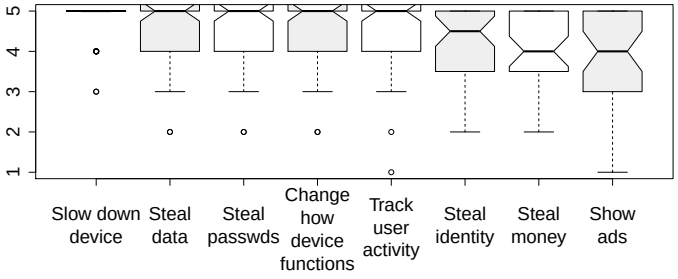


Fig. 2. What does malware do?

likely" or "strongly agree", and 3 for "neutral". The interquartile range of responses is indicated by the upper and lower boundaries of the boxes, and the median response is marked by the horizontal line in the middle of the box. The notches (indentations) in the middle of the boxes show an estimation of the 95% confidence interval surrounding the median. When the notches of two boxes do not overlap, it suggests that the two medians significantly differ. In the remainder of this section, we report the median Likert-scale response to each question, out of five.

In response to being asked where malware comes from (Fig. 1), participants strongly agreed that they could get it from installing software (5/5 on the Likert scale). They agreed that emails, social media, and public wi-fi were also sources of malware (4/5). Sharing devices with others was only moderately agreed upon as a source for malware (3.5/5), and using other people's devices was generally disregarded as a source of malware (2.5/5). People were unsure if sharing a password could introduce malware (3/5). Overall, these answers seem reasonable, reflecting the dangers typically emphasized in cybersecurity education.

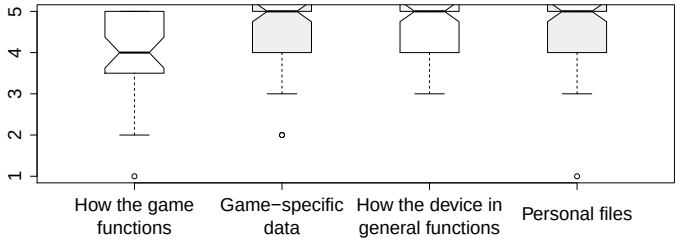


Fig. 3. If you installed a video game that has been affected by malware, what would it affect?

When asked what malware does, the answers were as follows (Fig. 2). Everyone strongly agreed that the malware would slow down your computer, track your online interactions, steal your passwords and data, and change the computer's functionality. They also agreed that malware can steal your identity information (4/5), money (4/5), and show ads (4/5). This also seems to indicate that users are aware that things, such as theft, can occur because of malware.

When participants were presented with a scenario where they had installed a malware-affected video game and asked what would be affected (Fig. 3), participants were in general strong agreement (5/5) that the malware could affect the game data as well as personal files and overall system functionality. They also agreed (4/5) that the game functionality would be affected as well. These answers suggest that participants are aware that malware is not confined to the specific pieces of software, but has the ability to affect the entire system.

In response to the question asking participants what items malware affects (Fig.4), most participants were sure (5/5) that malware could affect laptop and desktop computers, and files (4.5/5). Participants were also sure (4/5) that smartphones, software/apps, games, USB drives, social media, and smart watches could be affected by malware. Participants disagreed that malware affected DVDs/CDs (2.5/5), monitors (2/5), mice/keyboards (1.5/5), headphones and cables (1/5). These results are all reasonable.

Participants were less sure (3/5) of the affect of malware on home networks, smart appliances, and home alarms. This indicates that people do not have developed mental models of malware in the context of potential security flaws of smart

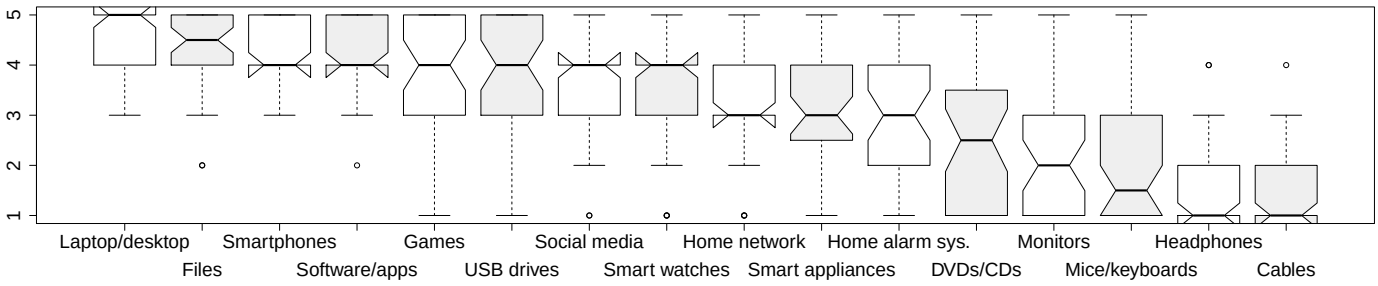


Fig. 4. What does malware affect?

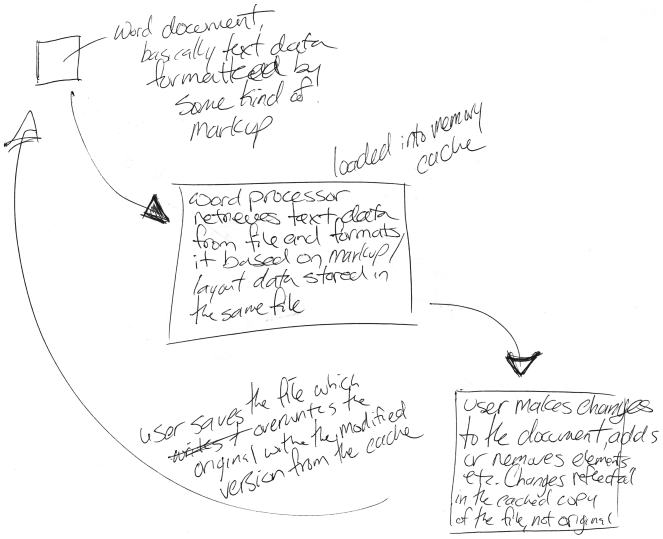


Fig. 5a. A1: Software is sequential transformation.

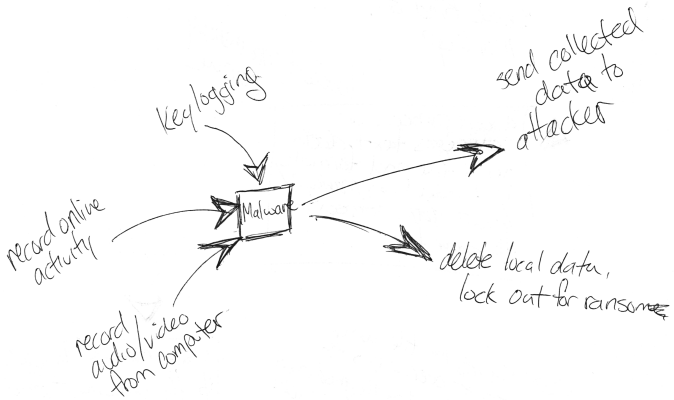


Fig. 5b. A2: Software is sequential transformation *with malicious results.*

devices and the Internet of Things.[2]

Our questionnaire targets *declarative* knowledge, which is factual knowledge—knowledge that something is the case. Knowledge of this type does not depend on any deeper understanding on how the larger system they are a part of works. With our diagramming exercise we sought to measure a more procedural and integrated form of knowledge—knowledge about *how* malware and regular software *work*.

B. Diagramming Exercise

With regard to the overall structure of the drawings, we observed a wide variety among participants. The most noteworthy difference between participants was the degree to which they situated the capabilities of the software in a sequence. Many, like Participant A (Figs. 5a & 5b), linked all elements together. Others, like Participant B (Figs. 6a & 6b), have very few links between the elements in their drawings. C1 (Fig. 7a) is representative of a common pattern where software is depicted as a simple list of its capabilities.

[2]https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html

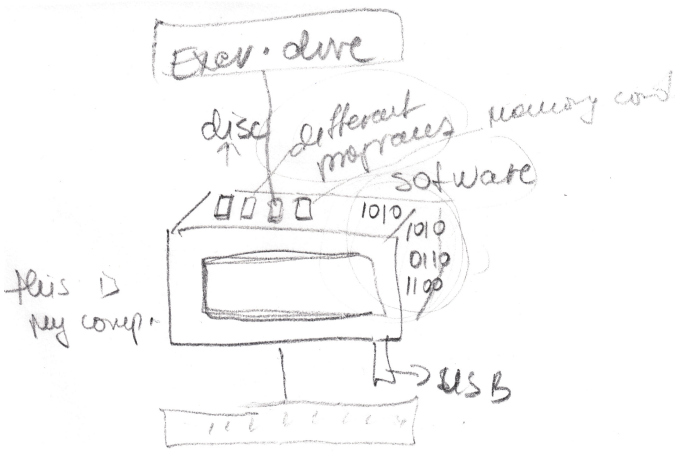


Fig. 6a. B1: Normally, a computer is structured and orderly.

Fig. 6b. B2: Malware is a chaotic infestation of this orderly system.

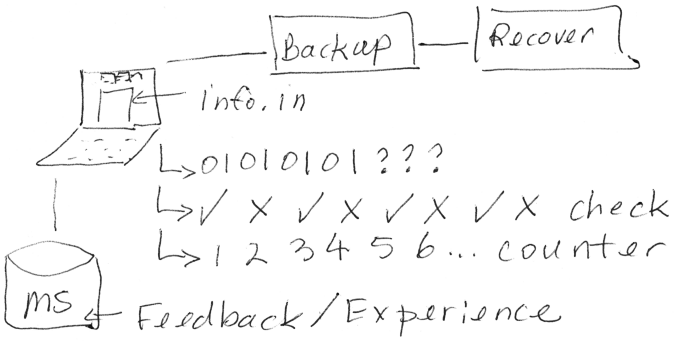


Fig. 7a. C1: Regular software is *good* capabilities.

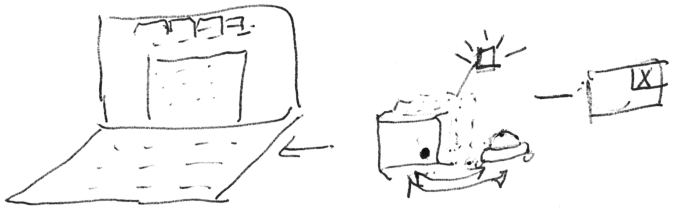


Fig. 7b. C2: Malware introduces new, *malicious* capabilities.

Observationally, there was a relationship between technical knowledge and drawing structure, with more knowledgeable participants tending to draw pictures with more complex structure.

Within participants, however, the structure of their two drawings were generally consistent. Drawings of malware tended to have less structure than drawings of software, which may reflect a lesser understanding of malware.

We performed open coding of the drawings of malware and regular software, in which we go through each picture and tag them at the lowest interesting level of meaning.

All participants emphasized the *capabilities* of both malware and regular software. Participants regarded malware as something that *eats*, *kills spreads*, *spies*, *exfiltrates*, *infects*, *infests corrupts*, *glitches*, and/or *slows*. After grouping these codes into higher-order units, we found that almost all participant drawings of malware could be placed in one of three major categories. Twenty-two out of 40 participants regard malware as something of a *nuisance*: malware is destructive to one's computer and/or data. Fourteen participants focus more on malware's *exploitative* potential: malware collects information on users, which can be used against them. Four participants think of malware as something that only *spreads*—to other devices, and to other parts of one's own device.

Our open codes of the word processor drawings revealed less variety in the drawings compared to our codes for malware. Users only depicted things that were directly related to UI elements and underlying processes that support the creation and editing of text documents.[3] Word processors *take user input* and *produce output*; they afford *opening*, *editing*, *styling*, and *storing* text documents; to achieve these capabilities, word processors depend on *underlying computational processes and computer hardware*. A typical drawing featured a monitor showing a Microsoft Word-like user interface showing key functionality such as styling, spell-checking, and saving; a keyboard for text entry; and a stick figure drawing of the primary user. Some participants opted to describe the hidden processes and hardware components that give rise to word processor functionality.

We were especially interested in how drawings of malware and regular software compare with one another. This comprised the 'axial coding' phase of our grounded theory analysis, which is the subject of the next three subsections. We identify two major contrasts in content between depictions of malware and regular software: *malware serves different ends than regular software*, and *malware is less known than regular software*. In the final subsection (IV-B3) we present our theory: the code that joins together these two higher-level codes: *regular software is controlled by the primary user, but malware is not*.

1) Malware serves different ends than regular software: Generally, malware is depicted as something that pursues different ends than regular software. Regular software obediently

[3]Participant C is a singular exception: they were the only participant to depict telemetry capabilities in their word processor drawing.

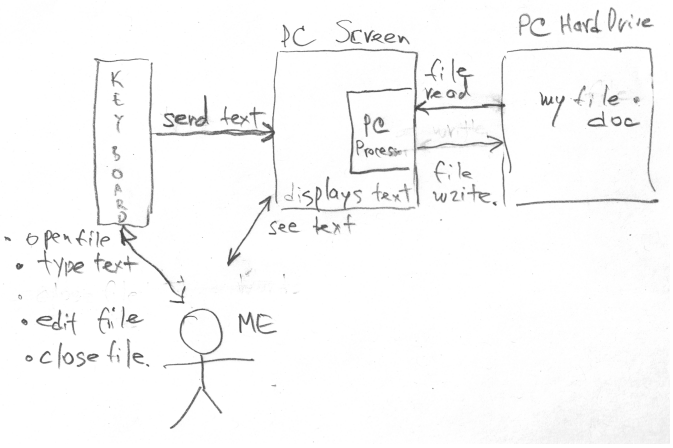


Fig. 8a. D1: Regular software is harmonious bi-directional communication between human and machine.

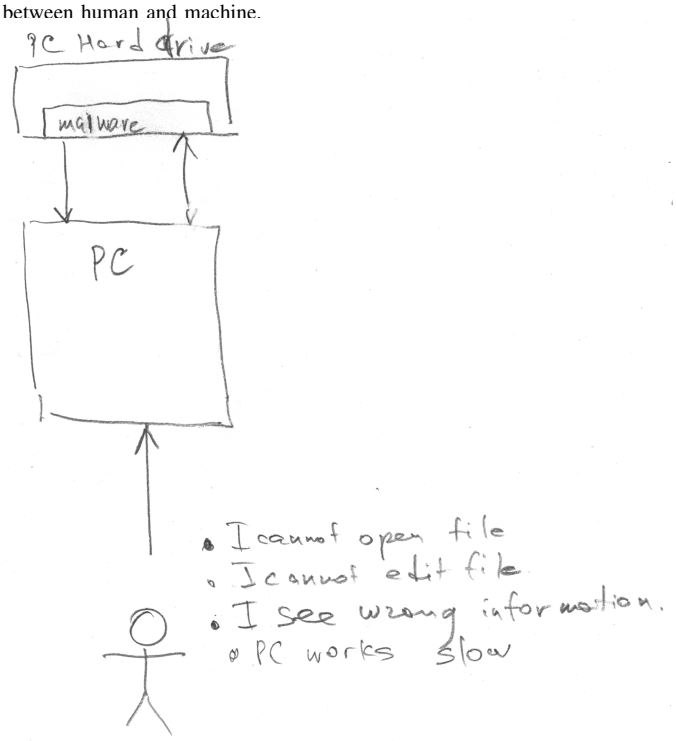


Fig. 8b. D2: Malware inserts itself between user and hardware, blocking user input and using hardware for its own purposes.

serves the end user and the end user alone, whereas malware either pursues its own ends, or the ends of an attacker.

For example, the two drawings of Participant A appear superficially similar, but there are major differences in their respective content. In A1 (Fig. 5a), there is a distinct focus on a certain type of data (i.e. a text document), and in describing how its state and contents are transformed throughout the process. In A2 (Fig. 5b), however, there is no focus on any particular type of data, and most details about how what malware does inside the computer is left out. The malware collects all types of data from a number of different sources, and either forwards it to an attacker, deletes it, or locks the

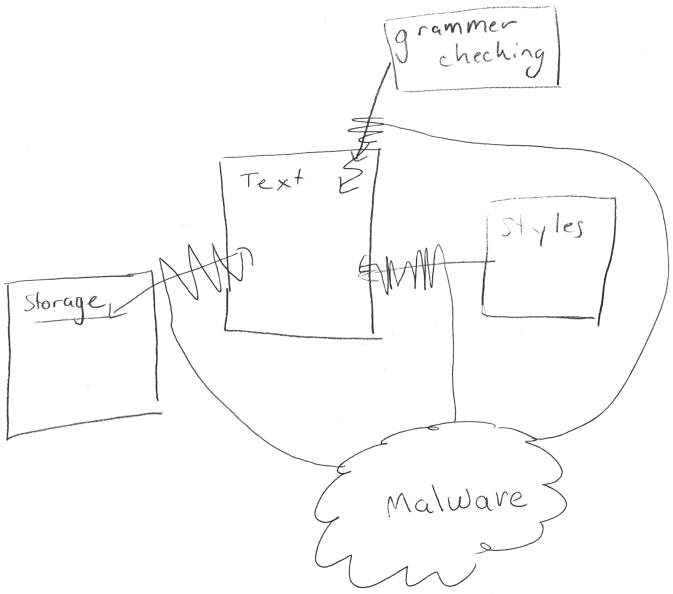


Fig. 9. E: Regular software is an integrated network of capabilities serving the user. Malware inserts itself between the modules, taking over the system.

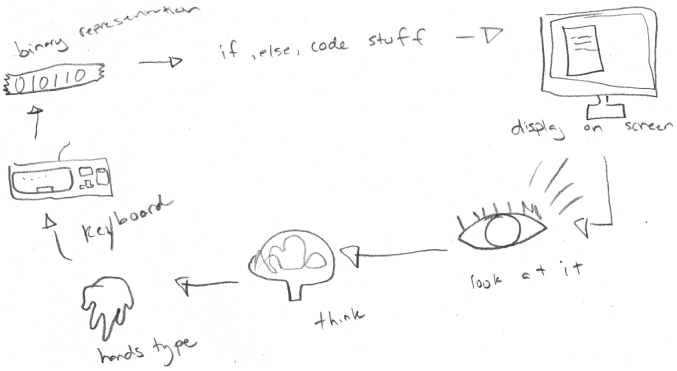


Fig. 10a. F1: Regular software is perfect input-output loop, where the user's thoughts are translated to a digital form and shown back to the user.

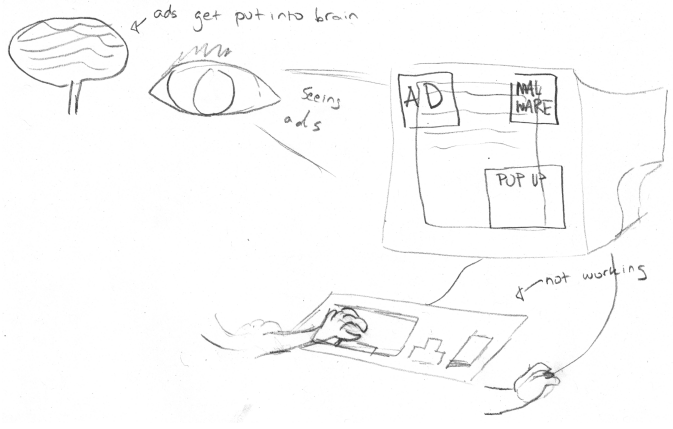
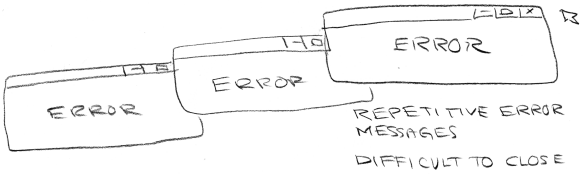


Fig. 10b. F2: Malware blocks user input and shows advertisements to the user.

Fig. 11a. G1: Regular software maps user input to various functions.



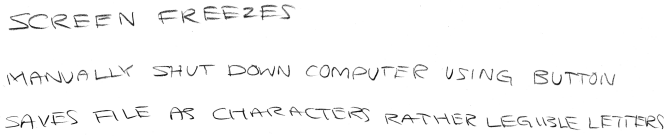


Fig. 11b. G2: Malware shows error messages and causes system malfunction.

user out of it to collect ransom.

Malware is frequently shown inserting itself inside the computer/software, disrupting harmonious human-machine interaction, often blocking-out the user in the process. In D1 (Fig. 8a) there is a bi-directional flow of information between the user and their peripheral devices, but in D2 (Fig. 8b)the user's inputs are no longer reciprocated. The malware has inserted itself between the user and the hardware, and it appears to be using the user's system for its own purposes. A similar depiction is shown more simply in the drawing by Participant E (Fig. 9), where the malware has interrupted and perhaps intercepted communication between several modules. In F1 (Fig. 10a) we see a perfect input-output loop where the users' thoughts are represented digitally and then displayed on the screen, but in F2 (Fig. 10b) the user's input is blocked, and instead of seeing the products of their imagination on screen they are shown advertisements on behalf of the malware. The pair of drawings by Participant G (Figs. 11a & 11b) are essentially simpler instances of the same general pattern seen in F1 and F2.

In fact, for most of the drawings of a word processor, participants were generally consistent in focusing on its positive, helpful attributes. Participant drawings of malware

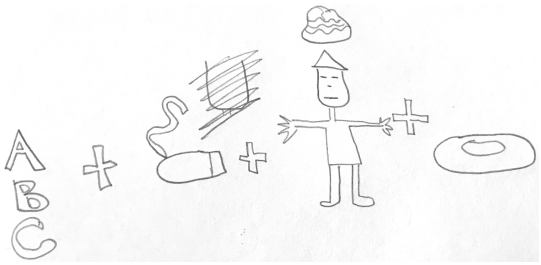


Fig. 12a. H1: Regular software is a robot butler.

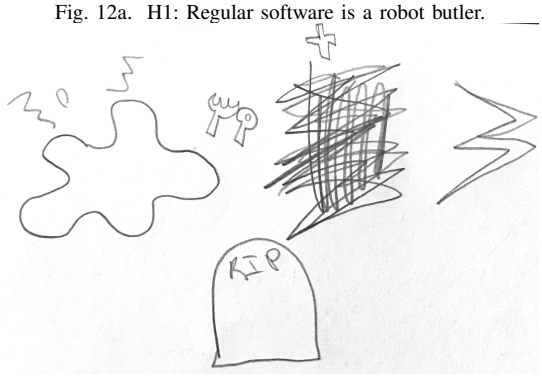


Fig. 12b. H2: Malware is a deadly pathogen.

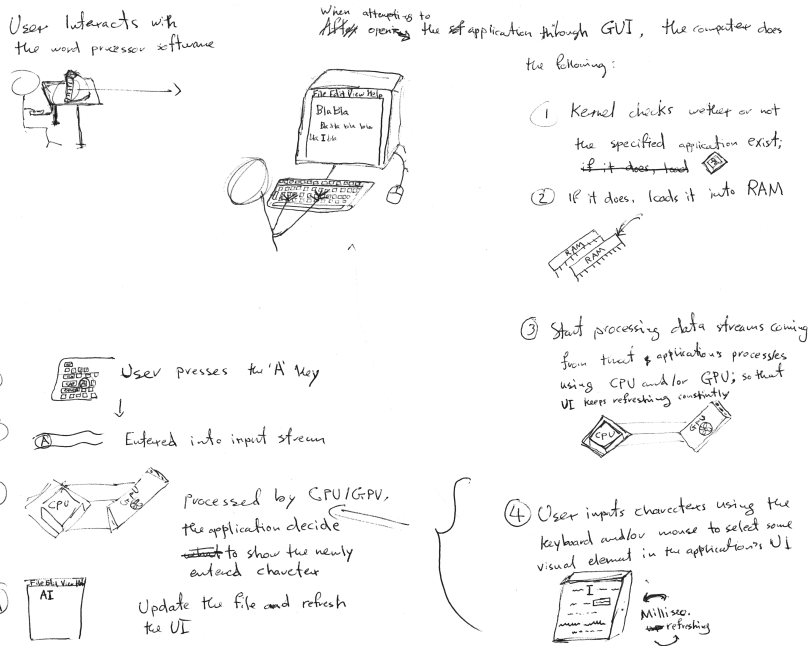


Fig. 13a. I1: Detailed description of a word processor

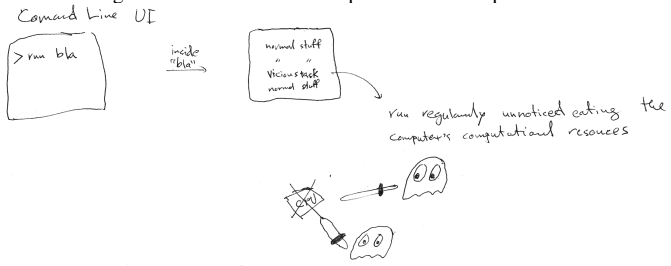


Fig. 13b. I2: Vague description of malware

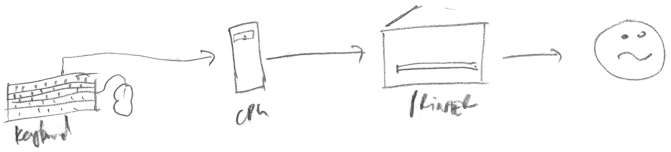


Fig. 14a. J1: Regular software is simple input and output serving the user.

Fig. 14b. J2: Malware eats data.

showed more heterogeneity, with malware being ascribed a wide variety of essential characteristics. Virtually all of these characteristics of malware were *wholly harmful* to the user, and their software and data, however[4]. For Participant C (Figs. 7a & 7b), malware is (a) a turtle which slows the device, (b) something that exfiltrates, and (c) something that produces pop-ups. Participant H (Figs. 12a & 12b) depicts regular software as a helpful robot, but malware as a pernicious virus-like organism seeking destruction. Participant B (Figs. 6a & 6b) depicts regular software as a highly ordered system, but the malware-affected word processor as a chaotic infestation of bugs who "delete…docs". Participant J's malware eats the user's software and data, G's (Fig. 11b) causes error messages, and D's blocks functionality and slows the device. And for many, malware is simply something that spreads: to other parts in one's own device, to one's files, and to other devices.

Overall, we found that **our participants view the two kinds of software as fundamentally different things**.

2) Malware is less known than regular software: In all but a few cases, participants' drawings of malware contained less detail than their drawings of a word processor, suggesting a lesser understanding of malware. This can be seen, for example, in the drawings of Participant B (Figs. 6a & 6b), Participant D (Figs. 8a & 8b), Participant G (Figs. 11a & 11b), Participant E (Fig. 9), and Participant A (Figs. 5a & 5b), where the drawings of malware have fewer distinct entities and fewer labels compared with the word processor counterparts. Perhaps the most striking example is in the drawings of Participant I, shown in Figs. 13a and 13b. This finding appears to apply to participants across the spectrum of technical knowledge—Participants A and I have some of the highest technical knowledge of the participants in our study.

3) Primary users control regular software, but not malware: These two codes just discussed are the "driving force" behind

[4]Participant I (Fig. 13b), is a rare exception: "vicious" tasks are executed alongside "normal" tasks.

our theory, which is that the concept of *control* is at the heart of participants' mental models of malware. Our participants unanimously thought of malware as insubordinate to the primary user. Regular software, by contrast, obeys only the primary user. Participants were also less detailed in their descriptions of malware than regular software, suggesting they know less about the former. The ability to control a system depends on knowledge of that system. These two themes thus represent two dimensions of control, one pertaining to perceived control, and the other to capacity for control. For our participants, regular software is something that is under their control, and malware is out of their control. We discuss some implications of these findings in the following section.

V. Discussion

A. RQ1: What are people's mental models of malware?

The results of the questionnaire suggest that people have reasonable factual knowledge of malware. Participants correctly identified *installing software*, *emails*, and *using public wi-fi* as likely sources of malware. Participants recognized the privacy and security ramifications of being affected by malware, stating that malware was likely to *track online interactions*, *steal passwords and data*, *steal identities* and *steal money*. The effects of malware were not restricted to a specific program or types of data: participants were aware that malware could *change how one's computer or device functions*, that malware could affect *personal files*, and that malware was likely to *slow down one's device*. A wide variety of devices were said to be potentially affected by malware: *laptop and desktop computers*, *files*, *smartphones*, *software/apps*, *games*, *USB drives*, and *smart watches*.

One issue of concern is that participants tended to rate smart watches, smart home appliances, and home networks as less likely to be affected by malware than things like laptop and desktop computers and smartphones. We see no clear evidence that the first group of devices are less likely to be affected by malware than a smartphone. Perhaps these responses are due to a relative unfamiliarity with these items. Smart watches and smart home appliances are new technologies that typical users may lack experience with, and we suspect that the majority of users do not interact with their routers in an administrative role. We suspect that increased familiarity with these devices would be accompanied by a greater appreciation for their susceptibility to malware.

Our participants demonstrate reasonable knowledge about malware's purposes, effects, and methods of transmission, and yet malware is still a major problem. However, as mentioned in the previous section, our questionnaire targeted *declarative*, (i.e. factual) knowledge of malware. According to the mental model theory, real understanding requires building models of how things *work*, which is not captured in our questionnaire. We therefore do not take these questionnaire results as evidence that participants' mental models adequately equip them for dealing with the dangers of malware. We think the diagramming exercise, which explicitly asked participants to show how malware (and regular software) *worked*, provide a better measure of participants' mental models.

When comparing drawings for malware with those for regular software, we found that participants tended to give their drawings of malware and regular software the same basic structure, but vastly different content. In the case of regular software, functionality was aligned with the user's goals of creating a text document. Malware, however, either had goals of its own, or was acting in service of a third party attacker. We also found that drawings of malware typically had far less information than their drawings of a regular word processor, suggesting that less is known about malware. For the remainder of this section we will refer to these two findings as malware *obscurity* and *autonomy*.

Overarching malware *autonomy* and *obscurity* is the concept of *control*. Regular software is controlled by the primary user, whereas malware is not. We believe that the concept of control is at the heart of our participants' mental models of malware. To our knowledge, this is a novel finding. In the remaining part of this section, we discuss some implications our findings have for decision-making related to malware.

B. RQ2: Are there characteristics of people's mental models which suggest how systems can be designed so that people are less susceptible to malware?

The finding of malware *obscurity* is important because attackers exploit gaps in users' knowledge of how malware works [31]. There may be an opportunity to reduce users' online vulnerability by improving user understanding of malware.

The findings of malware *autonomy* and *control* are both loosely accurate: malware serves someone other than the primary user, and malware is designed to elude our control. However, our analysis suggests that participants make a black-and-white distinction with regular software in these dimensions that we believe is unjustified, which has potentially negative consequences for security behaviour.

If users place malware and regular software in completely separate categories in terms of who they benefit, this may make them vulnerable to malware that behaves in some ways like helpful software. Malware can perform helpful functions in addition to functions that take advantage of the user. For example, illicit ("cracked") copies of paid software are often malware, though they may be superficially indistinguishable from the "real" software they imitate. Thinking of malware as wholly harmful and regular software as wholly helpful could result in unsafe behaviours such as freely installing and using any software that appears to be helpful. A more nuanced view of malware may make users more vigilant, and resistant to these kinds of attacks.

While users do have more control over regular software than malware, their drawings suggest that they are *fully* in control of regular software. This is inaccurate, and could be accompanied by a harmful overconfidence when it comes to software use.

To fully control a system, a complete understanding of it is necessary. Operating systems and the applications that run on them are too complex for anybody—even an expert—to fully understand. Modern user interfaces have made it possible for people of all technical backgrounds to enjoy the benefits of computers by hiding the underlying complexity through abstraction and encapsulation. UIs are so successful at this that they may even create the illusion of control. The feeling of control over or something is accompanied by a feeling of confidence that we know what the outcomes of interacting with it will be—how could anything unexpected happen? In the context of software use, this feeling of control over software could put users in the mindset that nothing bad will happen as a result of the actions they take: a false sense of security. We think it would be helpful to give users a better sense of the control they have (or do no have) over software, and better yet, to provide a way for them to increase the actual control they have over software through mental model development.

C. Future Work: Security Mental Model Builders in Software

We think that these two issues, and the overarching issue of control, indicate that users have inadequate working models of software. We suspect that abstraction and encapsulation in user interfaces have contributed to this. One builds an understanding of a system by building a representation of its "relation structure" [8]: the causal structure holding together its processes. In software, however, the underlying processes are largely hidden from users, and it takes technical expertise to be able to uncover them. Abstraction and encapsulation have made software much easier to use, but harder to learn about how it works.

To help users develop good working models of software, we recommend designing software that selectively exposes some important aspects of its functionality to give users a glimpse of how these systems really work, leading to better working models of software. In the case of trying to build mental models to improve behaviour related to malware, an emphasis should be placed on showing security-relevant aspects on software functionality. Of course, great care would have to be taken to ensure this information is provided frugally, in a way that would not overwhelm, confuse, or unduly interfere with users.

This addresses the problem of *control* by giving users a better appreciation of the degree of control (or lack thereof) they have over software, while empowering them to have more actual control. With richer understanding of the security-related aspects of software, users will be able to better understand the consequences of their decisions, the ways they are vulnerable, and possible countermeasures.

D. Limitations

We do not claim that our results are conclusive. An interesting pattern emerged in the data we collected, and we speculatively offer an explanation. Our results should be followed-up in future studies, which we plan to do.

Our diagramming exercise asked participants to draw only a Trojan. Ideally we would have asked them to draw several varieties of malware, but time constraints limited us to two drawings and the questionnaire. A comparison of drawings of multiple types of malware would make an interesting subject of future work.

VI. Conclusion

Users come to be affected by malware as a result of the decisions they make while using software. Mental models inform our expectations for the outcomes of interactions with software, so we set out to learn more about user's mental models of malware with the ultimate goal of helping improve these models so users can make better decisions. We designed and ran a study featuring a questionnaire on the purposes, effects, and methods of transmission of malware, and a drawing task where users illustrated their understanding of how malware and regular software worked. We found that participants have a reasonable factual understanding of malware, but some concerning patterns in their mental models of malware compared with regular software. Our participants showed a lesser understanding of malware compared to regular software, and made unjustified black-and-white distinctions between malware and regular software in terms of who the software benefits, and who is in control of the software. Our observations of software behaviour inform our mental models of it, so we suspect that this lack in mental models of malware and software is at least in part due to abstraction and encapsulation in user interfaces, which eliminate complexity to make software easier to use. For future work we recommend exposing some aspects of software functionality to allow users to develop better mental models of how they work.

Acknowledgments

We thank Paulina Chametka for her help with data collection.

References

[1] Ruba Abu-Salma, Angela M. Sasse, Joseph Bonneau, Anastasia Danilova, Alena Naiakshina, and Matthew Smith. Obstacles to the adoption of secure communication tools. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 137–153. IEEE, 5 2017.

[2] Farzaneh Asgharpour, Debin Liu, and L. Jean Camp. Mental models of security risks. In Sven Dietrich and Rachna Dhamija, editors, *Financial Cryptography and Data Security*, pages 367–377, Berlin, DE, 2007. Springer.

[3] Lawrence W. Barsalou. Perceptual symbol systems. *Behavioral and Brain Sciences*, 22(4):577–660, 1999.

[4] Cristian Bravo-Lillo, Lorrie Faith Cranor, Julie Downs, and Saranga Komanduri. Bridging the gap in computer security warnings: A mental model approach. *IEEE Symposium on Security and Privacy (SP)*, 9(2):18–26, 3 2011.

[5] L. Jean Camp. Mental models of privacy and security. *IEEE Technology and Society Magazine*, 28(3):37–46, 2009.

[6] Kathy Charmaz and Linda Liska Belgrave. Grounded theory. *The Blackwell Encyclopedia of Sociology*, 2007.

[7] Peter Pin-Shan Chen. English sentence structure and entity-relationship diagrams. *Information Sciences*, 29(2-3):127–149, 1983.

[8] Kenneth James Williams Craik. *The nature of explanation*. Cambridge University Press, Cambridge, UK, 1943.

[9] Clinton B. De Soto, Marvin London, and Stephen Handel. Social reasoning and spatial paralogic. *Journal of Personality and Social Psychology*, 2(4):513–521, 1965.

[10] Hans-Erik Eriksson and Magnus Penker. *Business modeling with UML: Business patterns at work*. John Wiley & Sons, Inc., 2000.

[11] Merideth Gattis. Space as a basis for abstract thought. In Merideth Gattis, editor, *Spatial Schemas and Abstract Thought*, pages 1–14. MIT Press, Cambridge, MA, 2001.

[12] Melvyn A. Goodale and David Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[13] Sander Greenland, Judea Pearl, and James M. Robins. Causal diagrams for epidemiologic research. *Epidemiology*, 10:37–48, 1999.

[14] John Brian Harley, David Woodward, and G. Malcolm Lewis. *The history of cartography*, volume 1. University of Chicago Press Chicago, 1987.

[15] Philip N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*. 6. Harvard University Press, Cambridge, MA, 1983.

[16] David Jonassen and Young Hoan Cho. Externalizing mental models with mindtools. In *Understanding Models for Learning and Instruction*, pages 145–159. Springer, 2008.

[17] Ruogu Kang, Laura Dabbish, Nathaniel Fruchter, and Sara Kiesler. My data just goes everywhere: User mental models of the internet and implications for privacy and security. In *Eleventh Symposium On Usable Privacy and Security (SOUPS) 2015)*, pages 39–52, 2015.

[18] David Kirsh and Paul Maglio. On distinguishing epistemic from pragmatic action. *Cognitive science*, 18(4):513–549, 1994.

[19] Predrag Klasnja, Sunny Consolvo, Jaeyeon Jung, Benjamin M. Greenstein, Louis LeGrand, Pauline Powledge, and David Wetherall. When I am on wi-fi, I am fearless: Privacy concerns & practices in everyday wi-fi use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1993–2002. ACM, 2009.

[20] Sanford Labovitz. Some observations on measurement and statistics. *Social Forces*, 46(2):151–160, 12 1967.

[21] George Lakoff and Mark Johnson. *Metaphors we live by*. University of Chicago Press, 2008.

[22] John Leach. Improving user security behaviour. *Computers & Security*, 22(8):685–692, 2003.

[23] Donald A. Norman. *The design of everyday things: Revised and expanded edition*. Basic Books, Inc., Hachette, NY, 2013.

[24] Donald A. Norman. Some observations on mental models. In *Mental models*, pages 15–22. Psychology Press, 2014.

[25] David Sinreich, Daniel Gopher, Shay Ben-Barak, Yariv Marmor, and Rakefet Lahat. Mental models as a practical tool in the engineer's toolbox. *International Journal of Production Research*, 43(14):2977–2996, 2005.

[26] David Stea, James M. Blaut, and Jennifer Stephens. Mapping as a cultural universal. In *The Construction of Cognitive Maps*, pages 345–360. Springer, 1996.

[27] Milica Stojmenović and Robert Biddle. Hide-and-seek with website identity information. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6. IEEE, 2018.

[28] Anselm Strauss and Juliet Corbin. *Basics of qualitative research*. Sage Publications, 1990.

[29] Barbara Tversky. Spatial schemas in depictions. In Merideth Gattis, editor, *Spatial Schemas and Abstract Thought*, pages 79–111. MIT Press, Cambridge, MA, 2001.

[30] United Nations Office on Drugs and Crime. *Criminal intelligence manual for analysts*. United Nations, Vienna, AT, 2011.

[31] Rick Wash. Folk models of home computer security. In *Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS) 2010*, pages 11:1–11:16, New York, NY, 2010. ACM.

[32] Rick Wash and Emilee Rader. Influencing mental models of security: A research agenda. In *Proceedings of the 2011 New Security Paradigms Workshop*, pages 57–66. ACM, 2011.

[33] Fern K. Willits, Gene L. Theodori, and A.E. Luloff. Another look at Likert scales. *Journal of Rural Social Sciences*, 31(3):126–139, 2016.

[34] Margaret Wilson. Six views of embodied cognition. *Psychonomic Bulletin & Review*, 9(4):625–636, 2002.

[35] Justin Wu and Daniel Zappala. When is a tree really a truck? Exploring mental models of encryption. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS) 2018)*, pages 395–409, 2018.

[36] Rolf A. Zwaan. Situation models, mental simulations, and abstract concepts in discourse comprehension. *Psychonomic Bulletin & Review*, 23(4):1028–1034, 2016.

[37] Rolf A. Zwaan and Gabriel A. Radvansky. Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2):162–185, 1998.