I just don't get it: Common Security Misconceptions

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Maz Jindeel

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Peter Peterson

June 2019

Acknowledgements

Thanks to Aleksandar Straumann, Jennie Smith, Brandon Geraci, and Peter Peterson for coding and all your contributions to this project. Thanks also to Abigail Pederson and Jennie for your work on the open-ended questions.

Dedication

To my family, roommates, and friends without whose support this wouldn't have been possible. Also to the Bogle family and R'hllor, who helped me through the long nights.

Abstract

Many security mistakes are made because of some underlying misconception about computer security. These misconceptions can be remedied by developing curricula targeting them, but they must first be identified. This paper outlines our process for identifying common security misconceptions by surveying experts and coding their responses and the results of that process. We also present open-ended questions which are preliminary version of a computer security concept inventory based on these misconceptions.

Contents

List of Figures

1 Introduction

Despite advances in computer science education, computer security mistakes remain quite common. Many of these mistakes are made because of an underlying misunderstanding about how security works. Subjects can be assessed regarding their understandings or base concepts, and these misconceptions can be remedied by creating a curriculum targeted towards them, but they must first be identified.

This goal of this work is to identify common security misconceptions by surveying experts about common security mistakes they've observed being made by novices, and coding those responses by underlying misconception in order to produce a list of misconceptions well-supported by data. A concept inventory will be created with questions to specifically assess each misconception. Finally, a curriculum will be developed to target these misconceptions with active learning modules and educational videos.

This paper describes the motivation for our study, the process of creating, using, and coding our survey, and our process of selecting and developing our list of misconceptions discovered through this process. We also discuss next steps and our vision for the future of this work.

2 Background

There are several integral concepts to understand for this work. First, it is important to consider survey design. We also have to understand how a coding process works and how it can be evaluated. Finally, we need to know how to develop a concept inventory based on a list of concepts. These concepts are discussed in this section.

2.1 Concept Inventories

A concept inventory is a test designed to evaluate a student's understanding of a set of concepts. Concept inventories are usually presented in the form of a multiple choice test. They can be administered post-course to verify students' understanding of concepts or both pre- and post-course to evaluate student learning. Distractors are based on common mistakes, usually derived from an open-ended version of the test. A score on a concept inventory should be a reflection of a student's mastery of whatever set of concepts are in the inventory.

The first concept inventory was the Force Concept Inventory (FCI), which tested students' understanding of Newtonian Physics[7]. Students held commonsense beliefs about physics which were incompatible with Newtonian Physics. Students with these beliefs failed to comprehend material in a physics course since they were learning on a faulty foundation of knowledge. For example, a student might hold the commonsense belief that heavier objects fall faster than lighter ones (even in a vacuum), based on

their observations of objects falling while subjected to air resistance. However, in a vacuum, objects of any weight or size will fall at the same rate. The Force Concept inventory forces students to make a choice between Newtonian and commonsense perspectives.

Concepts in a concept inventory are usually basic concepts in the field, not deeply complex and specific pieces of knowledge. Hestenes, et al. found that many professors considered the questions on the inventory too simple to be informative, but were surprised at their students' poor performance[7].

Almstrum, et al. outline a process for creating a concept inventory consisting of five steps[1]. The first step is determining the concepts that will be on the concept inventory. They suggest identifying concepts by surveying domain experts.

The second step is observing the process by which students misunderstand these concepts, typically done via interviews of either individual students or focus groups of students.

The third step is constructing the multiple-choice questions. This is usually an two step process. First, researchers generate open-ended questions focusing on individual concepts. Incorrect answers to these questions can be used to generate distractors and possibly provide insight into new misconceptions. In addition, students can be interviewed about why they believed their incorrect answers were correct. Then, data gathered from the open-ended version of the concept inventory is used to generate multiple choice questions .

The fourth step is the administration of a beta version of the concept inventory. This step allows researchers to study the reliability and validity of the test. Reliability is a measure of how consistently students will answer—how similarly will a student answer the same question if they took the test twice? Validity is a measure of whether the questions on the concept inventory reveal the misconceptions they are meant to

reveal. Reliability can be statistically tested in results of administration of the concept inventory.

Validity, on the other hand, is more difficult to evaluate. There are two types of validity to check—content validity and construct validity. Content validity can be verified by domain experts who review both the concept inventory's creation process and the inventory itself to see if it represents the domain well. Construct validity, on the other hand, is concerned with whether or not each item measures what it's intended to measure. This can be done in several ways, including statistical analysis showing answers to items about similar items are similar, analysis to correlate item scores with other measures, and careful expert review.

The fifth step is to iteratively improve the concept inventory.

2.2 Survey

Kelly, et al. out several components of planning a good survey-based study[10].

The first component is planning the topics of interest. They recommend collaborating with domain experts, colleagues, and members of the target population at this step in order to design good questions. Good questions should be properly formatted with proper capitalization, numbering, and question grouping with clear instructions. Double barrelled questions (i.e. asking two questions in one) should be avoided. Leading questions should also be avoided. The survey should also avoid having instructions that could bias responses.

Another component is piloting the study. In this step, beta testers should fill out the study and provide feedback, which can be incorporated into the final survey. The cover letter is another component, which should include some information about the organization behind the study (including information about the researcher), details

about how or why the interviewee was selected, goals of the study, and an explanation of how responses will be used.

Sample size is an important factor to consider when conducting a survey-based study. A qualitative survey typically needs less data than a quantitative study.

Malterud, et al. lay out a framework for evaluating necessary sample size of a qualitative study[11]. Their framework looks at a metric they call information power to calculate how large a sample is needed in a study. Information power depends on five factors—aim of the study, sample specificity, use of established theory, quality of dialogue, and analysis strategy. A study with a broad aim will need a larger sample size than a study with a narrow scope. Specificity refers to specificity of experiences, knowledge, or properties of the study's participants. A smaller sample size is needed where subjects have high specificity. Established theory asks about the level of theoretical background of the study. Studies supported by a well-established theoretical background require a smaller sample size than those that are supported by less established theoretical backgrounds. Quality of dialogue refers to how well interviewers and researchers communicate. Low quality dialogue leads to a higher sample size being needed, and visa versa. Analysis strategy can be either case-by-case or on a cross-case basis. Case-by-case analysis requires less participants than a cross-case examination.

2.3 Coding

Coding is a process by which objective results can be drawn from subjective survey results. A coding process was outlined by Popping (2015)[13]. There are two approaches to coding—one, the instrumental approach, approaches the data from the researcher's perspective during coding. The other—the representative approach—

considers the respondent's perspective when coding. Both of these perspectives—instrumental and representative—can be incorporated in a single coding process.

In Popping's process, coders independently generate codes for some subset of the results, then merge the generated codes for a final list of codes[13]. The coders then individually go over the entire data set, classifying each item using the list of codes generated in the previous step. There are some potential pitfalls to avoid when coding, such as including generating redundant, poorly defined, and/or non mutually exclusive codes[13]. Another issue is coders having different interpretations of the codes, an issue that can be somewhat alleviated during the generation of the final list of codes[3]. According to Hak and Bernts, coders should avoid socializing with regards to the coding as not to influence their coding[4].

It is important to evaluate inter-rater reliability. In other words, given the same data, what was the variance in how coders classified it? An agreement metric such as Cohen's Kappa (for two coders) or Fleiss's Kappa (for more than two coders) are recommended[13, 14].

3 Related Work

There have been several related works. Notably, several computer science and cybersecurity-related concept inventories have been developed. Additionally, some work has been done to identify common security advice and practices. We discuss some of these works in this section.

3.1 Cybersecurity Assessment Tools (CATS)

The Cybersecurity Assessment Tools (CATS) project aims to develop a set of tools intended to measure the quality of introductory cybersecurity courses[16]. One of these tools is the Cybersecurity Concept Inventory (CCI). The CCI is meant to evaluate students who have taken a single security class. The first step in their process for developing the CCI was to identify core security concepts through the use of a Delphi process, followed by student interviews to help understand why these misconceptions occurred. The CATS project is also working on a Cybersecurity Curriculum Assessment (CCA) project, meant for graduates of a security program. There were two simultaneous Delphi processes, one for the CCI and one for the CCA. Initially, the results of both Delphi processes were highly similar, so the CCI Delphi process was re-done with an emphasis on adversarial thinking. Thirty-six experts participated in the Delphi Process, all of whom had a PhD and were either working in or teaching cybersecurity. The restarted CCI Delphi process yielded 30 main areas of focus within cybersecurity, of which they focused on the top five. The five concepts the CCI

focuses on are “Identify vulnerabilities and failures”, “Identify attacks against CIA [Confidentiality Integrity Availability] triad and authentication”, “Devise a defense”, “Identify the security goals”, and “Identify potential targets and attackers”. Using twelve fictional scenarios as the interview basis, twenty-six student interviews were performed to see how students reasoned about different concepts.

The CCI draft was then generated. Each of the draft’s thirty questions was assigned to one of twelve fictional scenarios. These questions have only one correct answer, and distractors are inspired by the student interviews. Questions relate to a single concepts, although a single concept may be addressed by multiple questions. The process of validating the CCI is ongoing. A pilot study with 20 experts and at least 200 students is planned, the results of which will guide improvements to the CCI. They will then administer the CCI to at least 1000 students and analyze the results. Psychometric testing, cognitive interviews, and expert reviews will follow. The final test will likely consist of 25 questions—five questions for each of the top five areas.

At the time of this writing, the CCA is still in active development, but is nearing readiness for pilot testing.

3.2 Introductory Programming Concept Inventory

Caceffo, et al. developed a concept inventory for introductory programming using Alstrum, et al.’s method [2]. They first identified fundamental concepts. Then, through analysis of exams and instructor interviews, they identified misconceptions and potential distractors. In an iterative process, they produced open-ended questions, where students’ wrong answers led to better distractors and potentially new misconceptions. After several iterations, they will have enough distractors and mis-

conceptions to generate the final concept inventory as a multiple choice test. The proposed concept inventory consists of 21 multiple-choice and open-ended questions. As the concept inventory is still in the developmental phase, it has not yet been validated.

3.3 Digital Logic Concept Inventory

A digital logic concept inventory (DLCI) was developed by Herman et al.[5]. They used a previously compiled list of misconceptions, a survey of instructors, student interviews, and student responses to the alpha version of the DLCI to create the concept inventory. Base concepts to be evaluated by the DLCI were gathered from a group of instructors using a Delphi process. The DLCI contains multiple questions testing each concept in order to boost the test's reliability. Distractors each correspond to one student misconception (derived from student interviews and incorrect answers on the alpha version of the concept inventory). In some cases, where concepts only had one misconception, questions on the DLCI would address multiple misconceptions in order to have enough misconception-based distractors[5]. The DLCI consists of 15 items. In a follow-up paper, the authors present a framework for evaluating concept inventories and use this framework to verify the DLCI's validity and reliability[6].

3.4 Operating Systems Concept Inventory

A preliminary concept inventory for operating systems courses was created by Webb and Taylor[17]. Concepts were based on the authors' experience on what areas students tended to struggle in. Distractors were based on common student misconceptions. The concept inventory consists of ten questions and continues to

be refined. The authors propose statistical validation of the 10-question concept inventory after its completion.

3.5 Expert vs Non-Expert Security Advice and Practices

In addition to the various related concept inventories, there has also been some work investigating important security advice and knowledge. Done by researchers at Google, this work compares common security advice and the security practices of both experts and laypeople.

The Google researchers set out to answer two major questions. First, what advice would security experts give to non-experts? Second, what are the differences between the security practices of experts and non-experts? Lay people were considered non-experts, and people with 5+ years of computer security experience were considered experts.

To answer the first question, they surveyed 231 experts recruited via Google's Google Online Security Blog and through social media. Responses to an open-ended question "What are the top 3 pieces of advice you would give to a non-tech-savvy user to protect their security online?" are examined. A coding process was used to identify pieces of advice. First, the list of codes was generated. Next, the two coders both coded the same 10% subset of the data and examined their inter-rater reliability, achieving a Cohen's Kappa of 0.77. Next, the coders each independently coded half the data. The result was 837 codes assigned to each of the 231 responses (in some cases, more than three pieces of advice were identified per response). The 152 unique codes were then grouped by category and sub-divided by support. There was little

consensus among the experts on what advice was important and high priority[15].

The second question was answered with another survey question, which read "What are the 3 most important things you do to protect your security online?". This question was asked of the 231 experts and 294 non-experts on Amazon Mechanical Turk (MTurk). They found that the security practices of experts and non-experts differ significantly, and that while expert practices are generally in line with expert advice, non-expert practices are of mixed effectiveness[8].

They also outlined some guidelines for providing effective advice. Good advice should be effective (if followed, it should have the desired outcome), actionable (should be possible to do without being overly difficult), consistent (consistent with other advice and consistent in that the advice itself does not constantly change), concise (should be as small as possible). They acknowledge a trade-off between generic and specific advice. Generic advice tends to be more concise but can require skills and judgement that the advisee doesn't have, while specific advice might fail to address issues that are similar to the issue it addresses, but not exactly identical. For example, saying that vegetables are healthy is overly general—a diet of fried potatoes follows this advice but is not considered healthy. On the other hand, the advice that raw broccoli and kale are health is overly specific and does not mention that raw spinach is healthy as well. Some balance between general and specific advice is necessary. The two may be combined by first offering some generic advice followed by specific advice related to it for common situations.

4 Methods

At a high level, this work consisted of three parts. First, there was a survey, in which experts were asked to identify common security mistakes that they've observed. Second, there was a coding process by which misconceptions were derived from the mistakes experts reported. Third, there was the process of determining which misconceptions were highly supported and describing them.

4.1 Survey Design

The goal of our survey was to gather a list of common mistakes security experts observed being made b y novices. Two approaches were considered. The first was for the researchers to come up with a list of expected security misconceptions and have experts rate them by how common they were. The advantage of this approach is that it could have been easier for experts to rate items on a list than to come up with their own mistakes or misconceptions. The disadvantage of this approach was that we might leave out valuable mistakes that the experts had observed but that the researchers would not think of. The second approach was to ask researchers to come up with common security mistakes. This approach would allow each expert's experience to be represented in the final list of misconceptions, but it could be harder for experts to come up with mistakes they have observed. In the end, we chose the second approach so we could capture as much of each expert's experience as possible.

The survey started with a cover letter, explaining that the goals of the study, as

well as a description of what the study would be asking about. An example of a misconception from physics was used to avoid priming respondents with a security-related example. There was also an informed consent page where researchers were identified and information was provided about how their data would be used, how long it would be retained, and so on. This page also contained a box respondents had to check if they were willing to have their data included in the study. The confidentiality of responses was also assured—direct responses were not quoted anywhere in this study, nor will they released.

Designing appropriate questions was a challenge. On one hand, a single question such as "What are common computer security mistakes you have observed?" could be too vague and might generate only a fraction of the potential responses from each participant. An expert might think of a mistake they encounter daily, but might not have other mistakes come to mind in other areas of security. On the other hand, asking highly specific questions might shoehorn experts into specific fields instead of giving them the freedom to address mistakes in fields that weren't mentioned in the survey.

Another issue was that it's often easy for people to identify mistakes that they see, but can be difficult to generalize from a particular mistake to a broader misconception or pattern. We didn't want experts to be put off by that challenge. We decided it would be most valuable to have people identify as many security mistakes as they could and use our coding process to extract the underlying misconceptions. We also made it clear that the goal was to uncover misconceptions, and invited experts to articulate them on the survey if they could.

After considering the aforementioned issues, we decided to ask about six fundamental areas in computer security as a way to jog the imagination of our experts while staying fairly general. These areas are: "Network Security", "Application Secu-

rity", "Data Security and Encryption", "Physical Computer Security", "Information Privacy", and "Access Control". The security areas were adapted from Wikipedia's list of computer security areas[18]. We added an "Other" area as a catch-all in case experts saw common mistakes that they felt didn't fall under any of the categories. We asked several security experts if they felt any major categories were missing from the list, but they did not identify any additional areas.

For each category, we asked two open-ended questions, expecting answers in long text form. The first question was "Can you think of any security misconceptions or mistakes you have observed in $AREA?" and the second was "Why do you believe these misconceptions or mistakes occur?".

One way we tried to mitigate the difficulty of answering open-ended questions, as well as respecting our experts' time (and potentially their privacy), was to make every question on the survey optional. If experts did not have a mistake in mind for a particular area, they could simply leave it blank. The purpose of categories was to jog respondents' memory, not to sort their responses, and all responses were coded together.

We will describe our use of a coding process to analyze the results from the first question to identify the most commonly mentioned misconceptions later in this paper. The second question on the survey (Why do you believe these misconceptions or mistakes occur?) existed so that we could compare our results to our experts' intuition about root causes of the misconceptions, and to use them to flesh out descriptions of highly supported misconceptions and design the interventions.

Finally, there were some optional demographic questions. We asked for education level, which we split into nine categories (less than high school, some high school, high school diploma or GED, some undergraduate education, undergraduate diploma (2 or 4 year), some graduate school, master's degree (or equivalent), and PhD (or

equivalent), as well as an “other” section. We also asked ”What degrees or certifications do you have (if any)?” and ”What was/were your major(s)/minor(s) (if any)?”. The next question is ”In what sector do you work?” and has five options (Education, Industry, Government, Defense, and Other). Then we asked ”What is your primary job responsibility?”, which gives five 5 options (educating others, security research, applied security, consulting, and other). We also asked for how many years of experience in the field there are ”How many years of information security experience do you have?” with six categories (1-5, 6-10, 11-15, 16-20, 21-25, 25+). We also asked for an email address, if the participant is willing to receive follow-up communication about the study (”Please enter your email address if you’re willing to receive follow-up communication about this survey”).

Appendix A.1 contains the full text of the survey.

What sample size was needed for this survey? We analyzed the study using Malterud, et al.’s information power metric[11]. We saw that that our study had a broad aim, because we were asking about the entire field of computer security. Participants, who were specifically chosen because they are security experts exhibited high specificity. In terms of theory, while there have been a limited number of studies specifically looking at computer security misconceptions, there is a strong theoretical background for this coding process, concept inventories, and computer security. In this study, we did not consider dialogue quality in the same sense as Malterud, because there was not interview-style dialogue. However, because subjects could answer whatever they wanted, and questions were open-ended, we postulate that the quality of dialogue was rather high—if there was anything a subject wanted to mention, they were free to do so. Finally, our analysis strategy was cross-case, so we required more subjects. Practically speaking, how much data did we actually need? While the framework does not provide specific sample size recommendations, we know that

because we had high information power, we needed a relatively lowered sample size.

The survey was sent to around 40 security experts who volunteered to beta test it, of which around 10 responded. After accepting some minor revisions to demographic questions, the survey was hosted on through Google Forms and was left open for around two months, from September 2018 to November 2018. The survey was not substantially changed after the beta testing process, so the beta testing responses were included with the rest of the survey responses.

4.2 Finding Survey Experts

We went through conference proceedings for three security and security education conferences (Usenix, ASE, and 3GSE) and found names and email addresses of as many presenters as we could. As emails are not provided within the conference proceedings, this involved searching for names and affiliations and manually finding email addresses. In some cases, email addresses were listed directly on personal webpages. In other cases, we examined things like git commit history on GitHub accounts to identify email addresses. In total, we gathered the email addresses of 2500 academic experts. We also identified 254 industry experts from the Twitter information security community.

The 2500 academic experts were sent the survey via email. For the industry experts, 169 were contacted via email, and the rest were contacted via Twitter Direct Message. The survey was sent to a total of 2754 participants.

An identical yet separate survey was posted to several public security-related forums. Responses to this survey were kept separate from those to the emailed survey because it was not possible to verify the respondents were actually security experts, and to prevent any malicious entries from contaminating all the data. With one

exception (discussed later), we ended up combining the responses from both surveys in our analysis of the data.

4.3 Coding Process

Our coding process was based on the process outlined by Popping, with some variation [13]. In Popping's process, coders independently generate codes, merge their codes, then re-code the data using the newly generated codes. In our process, coders first generated their individual codes and independently coded the entire dataset. Then, a final list of codes was generated, which consisted of every individual code with a support count of three or more. As this work is only considering top misconceptions, three was chosen as a threshold where anything with a support count lower than three would not have been one of the most highly supported misconceptions. The codes on the final list were then merged so there was one final list of codes. Coders defined a mapping between their individual codes and final list codes. Coders evaluated codes with support counts of less than three to see if they would better map to a misconception on the final list and defined that mapping if they did. If not, these codes were deemed not well-supported enough to be considered. A script was then used to translate all the coding sheets to the final list of codes using these mappings. We consider misconceptions where a majority of the four coders had a support count $>= 3$ to be the most highly supported misconceptions we discovered.

Great care was taken by coders not to socialize with regards to the coding so the process was as unbiased as possible. Coders were each given their own copy of the data and instructed not to discuss the coding with others until it was finished. Coders also refrained from referencing lists or papers about common security mistakes so as not to influence the coding.

Inter-rater reliability was then assessed on the re-coded data using Fleiss's Kappa.

4.4 Creating Misconception List and Open-ended Stems

We wanted to create well-defined misconceptions. For each of the supported misconceptions from the coding process, we developed a concise statement of the misconception followed by a paragraph description of the misconception. The descriptions were developed by a group of seven researchers who individually worked on the descriptions and met weekly to review their work. Assigned descriptions were then rotated so researchers could work on all the descriptions. Open-ended stems were created using the same method of individually working on scenarios and questions, meeting to review, then rotating misconceptions.

5 Results

In this section, we discuss the results of the survey and coding process. Stastical validation of the concept inventory is also provided. This section also provides a review each of the highly supported misconceptions we discovered.

5.1 Survey Results

The expert survey received 75 responses, and the public forum survey received 14 responses. As all the public survey responses appeared reasonable (with the exception of one respondent, who reported no significant security experience, and whose responses were not considered), both the expert and public survey results (88 in total) were coded together. Each respondent had seven fields to enter data, corresponding to the questions in the survey. Discounting empty rows, we had 469 total data points.

In the expert survey, the majority of respondents (74%) reported having a PhD or equivalent degree. 8% reported having a masters degree or equivalent. The majority (68%) also worked in education, with 15% reporting working in the industry. 48% reported their primary responsibility as security research, 26% reported education as their primary responsibility, and 10% reported that their responsibility was in an applied security role. In terms of experience, there was a fairly normal distribution between all experience levels. No respondents reported an experience level of less than 1 year. 16% reported 1-5 years, 24% reported 6-10 years, 29% reported 11-15 years, 13% reported 16-20 years, 10% reported 21-25 years, and 9% reported having

over 25 years of security experience.

We received thirteen responses to the public survey. Of the respondents, 25% were reported having a Masters degree, and another 25% reported having "some undergraduate education". Another 16% reported having an undergraduate degree or "some graduate school". The remaining respondents either went to trade schools, vocational programs, or marked "N/A". 50% of the respondents worked in the industry, 20% worked in defense, 10% worked in Aerospace, 10% in Education, and 10% were not currently employed in a security position. Around half had some sort of extra certifications, such as CISSP, Comptia Sec+, CCNA, etc. Half of the respondents reported 1-5 years of security experience. A single respondent reported not having significant security experience. The responses for this respondent were not considered when coding the results of the survey and their responses are not referenced in figures or elsewhere in this paper. The remaining 42% of respondents to the public survey had over five years of computer security experience.

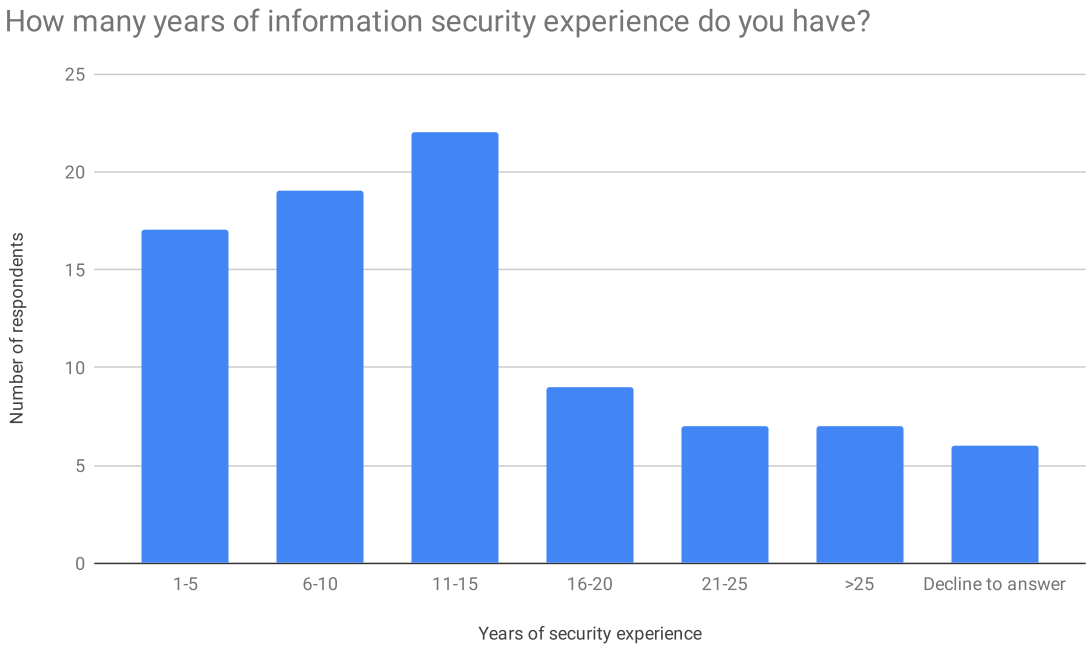
5.2 Coding Results

We ended up with 17 misconceptions supported by a majority of coders. Figure 5.2 shows the full list of misconceptions—supported misconceptions are those with a support count of three or more.

5.3 Statistical Validation of Coding Process

Fleiss's Kappa was used to evaluate inter-rater reliability of the coded data, finding k=.40, which is considered fair agreement.

Figure 5.1: Breakdown of survey participants by information security experience.



5.4 List of Supported Misconceptions

In this section, we provide a statement of each of our 17 highly supported misconceptions along with an explanation of the misconception.

5.4.1 As long as I'm using encryption, my data is secure.

This misconception is the belief that data is secure as long as encryption is being used. However, it fails to consider the limitations of encryption. For instance, data may be encrypted in transit but not while "at rest" (on storage media, a remote server, or in backups). It might not be clear to individuals how encryption schemes can be circumvented—for example, assuming that HTTPS is always secure, or that traffic is always secure when connected to a VPN. People who hold this misconception may also believe that cryptography automatically does things it simply doesn't, like guarantee

Figure 5.2: Full list of codes and their support counts

code	Coder 1	Coder 2	Coder 3	Coder 4	# of 0s	Count >= 3	Total	Text:
1	40	45	24	45	0	4	154	As long as I'm using encryption, my data is secure.
2	35	27	24	25	0	4	111	Physical security isn't as important as non-physical / technical security.
4	16	13	6	38	0	4	73	I am not a target of cyber attacks.
9	18	8	14	21	0	4	61	Following good password practices is not important.
6	12	11	6	16	0	4	45	This configuration works, so it's probably secure.
16	7	3	4	26	0	4	40	You can be completely anonymous on the internet by using privacy software and practices.
3	13	17	8	0	1	3	38	The software I use is secure, since the developers designed it with security in mind.
5	9	9	6	12	0	4	36	Having security product X makes me secure.
10	15	4	4	13	0	4	36	Humans are rational agents who understand security and can't be tricked.
11	15	0	6	14	1	3	35	I don't have to assign separate privilege levels because I can trust users to only do what they're supposed to.
8	7	6	9	12	0	4	34	Anonymized data can't leak sensitive information.
7	9	8	9	5	0	4	31	Keeping a processes secret is vital to its security.
18	18	3	5	3	0	4	29	Defense in depth is not necessary.
12	10	6	4	7	0	4	27	I can trust my users to not be malicious.
14	5	6	3	9	0	4	23	I have nothing to hide, so privacy isn't important to me.
21	5	7	5	6	0	4	23	Encryption automatically provides integrity and/or authenticity.
17	3	4	4	9	0	4	20	The inconvenience of Two Factor Authentication outweighs its security benefits.
27	0	8	11	0	2	2	19	it's not necessary / critical to encrypt data at rest / i don't need crypto
19	4	0	2	9	1	2	15	security should be baked in, not sprinkled on top
29	10	2	3	0	1	2	15	failure to consider insider threats
33	0	11	4	0	2	2	15	"strong passwords solve everything" / people use strong pws
20	0	4	10	0	2	2	14	I understand how to use crypto correctly and it's easy [or hard to get wrong] / including key security (pahp)
37	14	0	0	0	3	1	14	This small piece of data is not valuable on its own
13	6	2	5	0	1	2	13	you have explicit control over how information flows / is shared / leaks are forever
31	5	8	0	0	2	2	13	legal and policy protects my data / orgs care about my privacy
22	4	2	4	2	0	2	12	security needs technical solutions -- policies not critical
23	11	0	0	0	3	1	11	lack of logging / metrics / evalution / testing of security
24	7	0	0	3	2	2	10	making too complex / inconvenient (truth: leads to bad security)
25	4	1	3	0	1	2	8	what's my data worth / my data is not valuable / [scale of data collection - pahp]
15	0	0	7	0	3	1	7	misplaced trust (general mistake)
28	4	3	0	0	2	2	7	"I believe security is a binary thing - either you have it or you don't"
30	6	0	0	0	3	1	6	security can be achieved by a checklist (one size fits all)
48	2	0	3	0	2	1	5	Determining the security of something is a one-time decision (once secure, always secure). / "Fix-it-and-forget-it"
26	4	0	0	0	3	1	4	privacy is default on services I use / data must be explicitly shared
34	0	3	0	0	3	1	3	physical security can replace "computer" security
36	0	3	0	0	3	1	3	security is not my job / not important
38	3	0	0	0	3	1	3	Existing security policies and settings are based in logic and have good justifications

data integrity or authenticity. They might also fail to recognize that while encryption may hide the content of messages, in many cases it does not hide metadata about the communication or other externally-visible properties, such as the length of the message, the time between messages, similarities between portions of the messages, and so on.

5.4.2 Physical security is not as important as non-physical / technical security.

People often view security as a software problem with software solutions. However, physical security issues cannot be entirely solved by software – they commonly

require physical solutions (e.g., armed guards, locks, and tamper-resistant hardware) or responses. Ultimately, people who hold this misconception view physical security as *less* important than its software counterparts. As a result of this belief, they also believe that physical security breaches are less severe than software-based ones, but this is also untrue. Many software-based security solutions depend at least partially on physical security and can in many cases be bypassed with physical access. For example, password protections can often be circumvented by taking the hard drive out of computer and mounting it under a different OS, communication media can be compromised by installing a keylogger or network sniffer, keys and passwords can be acquired from individuals through threats or physical force, and content controls can be bypassed using cameras or other "analog logs".

5.4.3 I am not a target of cyber attacks.

People often think that since their individual data or information isn't "valuable" to anyone, no one would want to steal it. However, in the age of ransomware, people's data is as valuable to attackers as it is to the victims – as long as you'll pay to get it back, your data is a target. Similarly, some people seem to think that data is the only motive for a cyber attack. Instead, attackers may want to use a computer's resources, whether it's processing power for mining cryptocurrency or network bandwidth used for denial of service attacks or spamming, a computer's resources are valuable to attackers. Cyber attacks can also be automated, indiscriminate attacks that does not specifically target somebody, but victimize them nonetheless. Given these motivations, most people are likely victims of a cyber attack at some point[9].

5.4.4 Following good password practices is not important.

Victims of this misconception don't follow good password practices. They may not understand why their practices are insecure. For example, they may not understand that password reuse is bad, because if a reused password is leaked, all affected accounts will be vulnerable. They also may not understand what makes a password weak—passwords that would be "hard" for a human to guess may be easy for a machine to guess (whether they are weak to brute-force or to an attack via wordlist). They might never change their passwords, not recognizing that the longer a password is in use, the more likely it will be compromised. They might write passwords down in obvious places, not thinking that anyone would find them (or care to use them). The truth is that following good password practices can help keep accounts safe and are important to follow.

5.4.5 This configuration works, so it's probably secure.

Many people mistakenly conflate something working with it being secure. Software often has either default or example configurations that are easy to get working. These configurations are often not designed with security in mind and can include default passwords and credentials or leave security features disabled for ease of use. A related issue is that users may be aware that a configuration is insecure and intend to secure their system later, but they forget to do it. People might also not be aware of all the behaviors or options of software or devices that they're using, which can make it hard to fully secure them. In reality, it's important to remember that just because something works doesn't mean it's secure.

5.4.6 You can be completely anonymous on the internet by using privacy software and practices.

People holding this misconception put too much faith in the ability of privacy tools, and often do not understand their fundamental limitations. In reality, while these tools can increase your level of privacy, there is no way to be certain that you are anonymous on the internet. For example, The Onion Router (TOR) is considered the gold standard for anonymous internet browsing. However, if an attacker can control both exit nodes a user is using, they can tell what the user is browsing. While these attacks are considered theoretical, they highlight the fact that no privacy solution is guaranteed to always work.

5.4.7 The software I use is secure, since the developers designed it with security in mind.

People who hold this misconception put too much faith in software producers, assuming that those producers thought about security and did a good job building it into their product. Unfortunately, software is not always designed with security in mind, due to developers' lack of security awareness, ability, economic constraints, or other reasons. Additionally, developers might have carefully considered security, but simply made mistakes in the design or implementation phase. Or, they might have done a good job, but other conditions change resulting in security vulnerabilities (e.g., Meltdown). Novice software developers might even have this misconception about their own software, because they *tried* to create it to be secure. Whatever the reason, it is always a mistake to assume that software is secure.

5.4.8 Having security product X makes me secure.

Some users believe having a certain security product, such as anti-virus software, firewalls, or other security solutions makes them perfectly secure. Feeling secure, they might participate in risky activities, such as downloading and executing software from untrustworthy sources, thinking that their security controls will keep them completely safe. In reality, these products are often imperfect and always have to be used as part of a holistic security solution consisting of a combination of security defenses, policies, and practices with the knowledge that no security solution is perfect.

5.4.9 Humans are rational agents who understand security and can't be tricked.

This misconception is people making assumptions about themselves or others, such as "I won't fall victim to X scam or social engineering attack". People also make these assumptions when designing security policies, assuming people will act rationally, expertly and/or according to policy. Among other factors, inconvenience, time pressure, or authority structures can cause people to circumvent or disobey security policies. People might decide not to verify something and just assume it's OK, either because it's inconvenient to verify or because their attention is drawn elsewhere. In contrast, a realistic view of the limitations, weaknesses, and pressures facing ourselves and others is critical for building security mechanisms and policies that work in reality.

5.4.10 I don't have to assign separate privilege levels because I can trust people to only do what they're supposed to.

People who believe this misconception think that trust in users (e.g., local people in an organization and/or remote users of a program or system) or components of a system is a valid replacement for robust security controls. However, this idea breaks down whenever any person takes an action that violates this assumption, either intentionally (they are a bad actor) or unintentionally (they make a mistake). In essence, this misconception is not following the principle of least privilege, which says that people should get no more than the minimum permissions necessary to accomplish their tasks. Taking away any additional permission should break the system. A good way of building policies like this is to start from no permission ("deny by default") and then add only necessary permissions (privileges, features, etc.) until the system works as required. Following this principle can avoid both malicious unauthorized access and accidents.

5.4.11 Anonymized data can't leak sensitive information.

People who hold this misconception believe that anonymized data can't leak any sensitive information. However, data can be aggregated to reveal patterns or cross-referenced with other sources of information to de-anonymize individual records or reveal data about a population. For example, Netflix released a dataset for development of a better video recommendation algorithm. While the data was anonymized, it contained user reviews which cross-referenced with public movie review databases to de-anonymize those users[12].

5.4.12 Keeping a processes secret is vital to its security.

Victims of this misconception believe security by obscurity is a strong strategy. It is better to use a public process that's robust against attacks. This does not mean there cannot be any secrets in the process. Rather, it means that the secrets are separated from the process, so someone observing the process won't be able to get the secrets. For example, writing a "secret" encryption algorithm and using that along with keys is not as secure a process as using a public and robust algorithm and keeping the keys separate because the are two secrets in the secret encryption algorithm scheme—one is the actual key and the other is the encryption algorithm itself. In the scheme using a public and robust algorithm, there is only one secret—the key itself.

5.4.13 Defense in depth is not necessary.

People who hold this misconception think that a single defense is sufficient. However, relying on a single perimeter (or other) defense makes that defense a single point of failure. Since nothing is completely secure, it's possible that defense will fail, in which case there will be no other security mechanisms. In contrast, defense in depth means having multiple overlapping layers of security. By having defense in depth, a single layer defense failing doesn't mean the whole system is compromised – hopefully an additional defense will still provide protection.

5.4.14 I can trust my users to not be malicious.

People who hold this misconception trust that users or other "upstream" data sources will only provide harmless input. Users—either purposely or just through chance—may provide input that breaks the expected flow of the program. However,

instead of trusting input, if input is treated as malicious "until proven innocent" by validating and sanitizing it before use, dangerous input cannot cause bad outcomes. Additionally, because of this process, normal users with unusual input are likely to get a better experience, as the program will always perform as expected.

5.4.15 I have nothing to hide, so privacy isn't important to me.

What people who suffer this misconception fail to realize is that by giving up the right to privacy now, they are surrendering that right in the future, if they ever need or want it. They may also fail to realize that, by devaluing their own privacy, they may be making it harder for others to keep theirs. Individuals often do have information they would like to keep private, although they might not think about it or be aware of it. Data can also be in a different form than users typically identify with as personal information, such as values, interests, and other indirect information. Users fail to realize the importance of indirect personal information, which can be used to target advertising at them without their explicit consent.

5.4.16 Encryption automatically provides integrity and/or authenticity.

People who hold this misconception believe that encryption automatically provides integrity or authenticity along with confidentiality, while this is not always the case. Symmetric encryption algorithms used for confidentiality, such as AES, do provide integrity or authenticity in their most basic modes of operation. Authenticity is not provided because anyone with the AES key (which must be shared by its very nature) can send an encrypted message, and you cannot verify who sent it. Integrity is not

provided because a message's ciphertext can be modified in transit; decrypting it will result in data being produced which is not the plaintext, but no mechanism will identify this change. Instead, people needing authenticity and/or integrity should use cryptography designed for this purpose.

5.4.17 The inconvenience of Two Factor Authentication outweighs its security benefits.

People who hold this misconception see Two Factor Authentication (2FA) as not being worthwhile, possibly because they do not understand how, and the extent to which, it makes attacks against authentication harder. They also might underestimate the value of their data or their likelihood of being a target, which may make the hassle of 2FA seem unnecessary and wasteful. Two factor authentication may be inconvenient, but it provides a much stronger form of authentication by requiring both something the user knows (usually their login credentials of username and password) and something the user has (usually a cell phone or device). While it may be easy to steal one of the two factors, it's simply much harder to steal both. It's also impractical to steal both factors on a large scale (you might be able to steal a database with millions of passwords, but it's not possible to also steal all those users' phones). If we recognize that 2FA makes authentication attacks meaningfully harder, and we recognize the value of our data and access, then 2FA is a strategy worth considering very carefully.

6 Discussion

Our findings appear reasonable at face value. Retrospectively, however, there are several things we would try to improve if doing similar work. In particular, we would try to improve the coding process and consider our survey expert selection process and questions. These retrospective observations are discussed in this section.

6.1 Coding Process

Our coding process was non-standard in that each coder generated their own list of codes and coded all the data, then came up with a standardized list of codes. This approach was problematic in that different coders defined a different number of codes, and some split up a concept where others had more overarching concepts as one code. This deviation from standard coding practices was likely a contributing factor in our relatively low inter-rater reliability, as lower-supported misconceptions were often only supported by a single coder. We believe our results are still strong despite this reliability metric, especially when looking at the similar support counts all coders had for the top supported misconceptions.

Coders were given vague instructions at the start of the coding process with the intent of reducing the influence of instructions on their final coding. However, these instructions were inadequate, as one coder performed the coding in such a way that it was incompatible with the rest of the coding, and their coding were not considered as a result. Their codes generally supported our findings, but were not used in this

study.

Our coding process also did not include any student interviews when coming up with the codes or their descriptions. However, student interviews are planned with students who take an alpha version of the concept inventory.

6.2 Survey

Survey questions can often influence responses in some way, and this may have happened in this study. Our top two misconceptions concern people fundamentally misunderstanding the limitations of cryptography and people not considering physical security. Two survey questions specifically asked about these fields (cryptography and physical security), and many of the responses can be summarized as "they didn't consider it". It's likely that having any categorical questions would have had this same effect, which was something we didn't consider when trying to jog expert's memories of common security mistakes. We believe these misconceptions are highly important, so while they can be correlated closely with the survey questions, it not certain they were directly caused by the survey questions and it does not seem unreasonable that they are the top two.

The low response rate may be due to our method of selecting academic experts. We tried to find contact information for all the authors from the conferences mentioned above. However, many of these authors may have been graduate students when the papers were published, and might no longer be checking those email addresses or even working in the security field. A more narrowly targeted approach may have yielded a better response rate and more responses even if it did not find contact information for so many experts.

The low response rate might also have been improved by changing the survey.

While the survey was designed with a balance between highly specific and overly broad questions in mind, it may have been possible to jog people's memory while only asking a single open-ended question. The question might have asked "What are some common security mistakes you have observed in your work? Some areas of computer security to jog your memory are network security, application security, data security and encryption, physical security, information privacy, and access control. Please write down as many common mistakes you can think of, whether or not they fall into the above categories". This would leave only one question for respondents to answer, which may make them more willing to complete the survey. On the other hand, some of the survey respondents complained that the questions were too vague and difficult to answer. Further reducing the number of questions might cause more participants to encounter this issue. Including all the prompts above that one question might help with this issue, but there is not a clear best approach for asking these questions.

The only demographic question whose responses we considered in this part of the project was the years of security experience. The other questions may be considered in later parts of this project. However, if they end up not being considered, it would be good to remove such unnecessary questions from the survey in the future to make the survey as short and easy to take as possible. Additionally, a number of respondents worried that their responses to the demographic questions could be used to de-anonymize their responses. These questions were all optional, but removing them entirely would also alleviate this concern.

6.3 Coding Results

It is important to note that our process did not produce an objective, comprehensive list of the most common security misconceptions. Rather, we have taken a

snapshot in time of the experiences of 88 security experts. If this same work was repeated with different experts, we might see a somewhat different set of misconceptions on the final list. The coding process is also influenced to some degree by the coders own experiences. The coding process is a mixture of both hard and soft science, not an entirely objective process. However, it is a good method for giving some objectivity to purely subjective data (the survey responses). While some of the misconceptions may change if the work was repeated, a valid concept inventory can be created from this data, because the concept inventory is not a comprehensive test of the field. Rather, it is a sampling of common concepts analogous to a final exam in a class, where not all the concepts are covered.

This work was concerned with finding the most common security misconceptions. These are the list of top supported misconceptions we found. However, many of the misconceptions that did not make that list are still interesting misconceptions that are worth considering. For instance, one such misconception is the idea that security can be sprinkled in after a product is developed, instead of being part of the design. Another interesting misconception is that people fail to consider insider threats. The idea that an overly complex security system is more secure than one that is more simple is another noteworthy thing that was mentioned, but not highly supported in our data.

Some of the misconceptions that ended up being highly supported are somewhat surprising, like the last one about Two Factor Authentication. As a whole, however, the list of misconceptions appears to be a reasonable list. This list will be sent to some survey respondents who volunteered to look over the list of misconceptions for face validity in future work related to this project.

7 Future Work

A concept inventory consisting of the top supported misconceptions is being developed and is in the alpha phase, having been given to around 40 outgoing computer security students at the University of Minnesota Duluth. The concept inventory currently consists of 20 open-ended stems, and student responses will be used to create distractors and develop future versions of the inventory. This inventory will eventually be a validated instrument used for identifying students' understanding of the top misconceptions identified as part of this project.

A curriculum, consisting of educational videos and lab exercises is also being developed to target common misconceptions.

For each misconception, a hands-on or interactive activity will be created that demonstrates how the misconception is wrong and the right way of thinking about the misconception. These exercises will be for in-class or homework use and will include activity manuals and all supporting materials. These exercises will be available online and instructors will be able to choose which exercises are appropriate to assign to their students.

Videos will visually describe the misconception, why it is the wrong way of thinking, and the proper way of thinking about it. Videos will include examples of the misconception, whether they are real-world examples or a theoretical explanation. These videos will be publicly available and will be supplementary material for the lab exercises, but will not specifically describe the lab exercises, and instead focus on describing the misconception more generally.

8 Conclusions

Addressing common security misconceptions and developing educational tools to remedy them will improve the state of computer security. This work identifies common security misconceptions by surveying experts and coding the responses. Using 38 total codes, four coders coded the responses of the 88 survey respondents, ending up with a list of 17 highly supported misconceptions.

For each misconception, we have written a succinct statement of the misconception along with an explanation of the misconception and the correct way of thinking. At least one open-ended question was developed for each misconception to gather incorrect understandings of concepts to guide the future concept inventory.

This work is the first step in creating a misconception-based concept inventory and curriculum for computer security. Future steps will include the completion and validation of the concept inventory, along with the development of exercises and educational videos to accompany them.

A Appendix

A.1 Full Survey Text

Fundamental Computer Security Misconceptions

We are conducting research (NSF #1821788) to identify what security experts believe are significant, pernicious, important, or intuitive fundamental misconceptions held or mistakes made by security novices. A classic example from physics is that people commonly believe that a bowling ball will fall faster than a baseball, because it is heavier. In reality, gravity doesn't work that way, and reasoning based on that model will result in errors.

What we want to know is: What are the analogous beliefs that people have about computer security? Once we have a list of these misconceptions, we will be creating a test of those concepts along with active learning exercises to teach them to beginning security students. That's where you come in. In the context of seven areas of security – networking, applications, data security / encryption, physical security [as it relates to information security], privacy, access control, and a catch-all "other" area – we'd like you to describe what you have observed as common or critical misconceptions or mistakes in that area, and, if you can, describe why you think people hold those misconceptions or make those mistakes.

The survey ends with a short set of (optional) demographic questions about experience, employment and education.

Please note:

- The "areas of security" are only meant only to help you think broadly about security. Your responses will not be sorted by area, so if the prompts are unhelpful to you, feel free to put your answers under "other" or disregard the areas.
- Multiple answers per area are welcome.
- You are encouraged to leave fields blank according to your preference; any response will be helpful.
- For more information about our survey design, please see:

https://secmisco.blogspot.com/2018/11/survey-design.html

*** Required** Informed Consent and IRB Information Investigator: Dr. Peter A. H. Peterson – pahp@d.umn.edu

Eligibility: Participants must be at least 18 years old.

Overall Description of Participation:

It is your choice to take this survey or not, you can stop at any time, and you can skip any question on the survey you cannot or do not wish to answer. If you choose to participate, you will take a short survey (5-20 minutes depending on how much you choose to write) where you describe misconceptions about computer security that you believe are held by computer security novices. The survey asks the same question about misconceptions in several areas of security (and a catch-all "other" area). Following this, there is a short demographics survey about security expertise. This is NOT a test of your expertise in security. Again, you are encouraged to skip any question you cannot or do not wish to answer.

What will this data be used for?

We will use this data to create a list of common misconceptions. Based on that list,

we will create a test covering a set of those misconceptions, and hands-on exercises and videos targeting the most common or important misconceptions. We will publish anonymized and aggregate results to help the security community, and our educational materials will be made available for free online. Risks and Benefits of Participation:

No unusual risks or discomfort are expected from taking this survey (although some people find it challenging to articulate generic misconceptions in computer security). Benefits of participating may include enjoying the process of identifying misconceptions, and good feelings about contributing to research to improve security education and, hopefully, to improve future computer security.

Confidentiality and Data Use Statement: Survey responses to this Google Form will be stored in a Google Doc nominally accessible only to the project team. Your responses will be treated as confidential. Data will only be released in an aggregate rewritten form; your own words will not be released without further written permission from you. If you choose to provide your email address, it will only be used for follow-up questions relating to the survey (e.g., to ask for clarification or to ask to use your words verbatim). Your email address will not be released in any form. All email addresses in the survey results will be deleted from the stored data after the project is over. Responses will not be aggregated by respondent, but aggregated with other similar misconceptions in order to identify the most commonly mentioned misconceptions.

IRB Approval / Exemption: This portion of our project has been deemed by the UMN IRB as not involving "human subjects," and thus does not require approval OR an exemption, because, while it is 1) systematic research 2) meant to be generalizeable, it does NOT 3) involve personal AND identifiable information.

Check all that apply.

1. I consent to my responses being used as part of this study.
2. What are one or more of the most common and important misconceptions or mistakes in network security?
3. Why do you believe these misconceptions or mistakes occur?
4. What are one or more of the most common and important misconceptions or mistakes in application security?
5. Why do you believe these misconceptions or mistakes occur?
6. What are one or more of the most common and important misconceptions or mistakes in data security and encryption?
7. Why do you believe these misconceptions or mistakes occur?
8. What are one or more of the most common and important misconceptions or mistakes in physical security (as it relates to information security)?
9. Why do you believe these misconceptions or mistakes occur?
10. What are one or more of the most common and important misconceptions or mistakes in information privacy?
11. Why do you believe these misconceptions or mistakes occur?
12. What are one or more of the most common and important misconceptions or mistakes in access control?
13. Why do you believe these misconceptions or mistakes occur?
14. Can you think of any other security misconceptions or mistakes you have observed (in any area of computer security)?

15. Why do you believe these misconceptions or mistakes occur?

Demographics and Experience (optional) Information security is a field populated by people with a wide variety of experiences, employment and educational backgrounds. This optional demographic information will help us understand the backgrounds and expertise of respondents.

16. Job title

17. What is your highest level of education? Mark only one oval. Less than high school Some high school High school diploma or GED Some undergraduate education Undergraduate diploma (2 or 4-year) Some graduate school Master's degree (or equivalent) PhD (or equivalent) Other:

18. What degrees or certifications do you have (if any)?

19. What was/were your major(s)/minor(s) (if any)?

20. In what sector do you work, primarily? Mark only one oval. Education Industry Government Defense Other:

21. What is your primary job responsibility? Mark only one oval. Educating others Security research Applied security Consulting Other:

22. How many years of information security experience do you have? Mark only one oval. <1 1-5 6-10 11-15 16-20 21-25 >25 I do not have meaningful security experience.

23. Please enter your email address if you're willing to receive follow-up communication about this survey.

Thank you!

We really appreciate you taking the time to help us by providing your responses. We'll do our best to make the substantive results from this study public, and to create educational tools and materials to help remediate these important misconceptions.

For more information about our project, please see: https://secmisco.blogspot.com/ If you have any questions, concerns, or comments about this survey, please contact: Dr. Peter A. H. Peterson at pahp@d.umn.edu.

Thanks again!

A.2 Open-ended Stems

1. Suppose you're visiting a website and there is a signal in the interface (e.g., a green lock in the address bar) that signifies an encrypted connection. What is one reason why the connection might not be secure, even though it is encrypted? [5.4.1]

2. A facility stores extremely sensitive data on servers that are disconnected from the Internet. They are developing a new security policy. Given that the computers are disconnected from the Internet, and knowing that attacks exist to target air-gapped systems like this, what is the most critical component of the security policy? [5.4.2]

3. What makes someone a poor target for a social engineering attack (a security-related scam)? [5.4.3]

4. Alice determines that a random, 16-character password will take decades to crack. As a result, Alice creates one such password and uses it for all of her accounts. Explain why this is a bad strategy. [5.4.4]

5. Your neighbors have been stealing your WiFi, so you purchased and installed a brand new WiFi router from a well-known company for your home. After plugging it in, you're able to use the WiFi to access to the Internet. Why is it unlikely that this will solve your problem? [5.4.5]

6. Operating Systems are supposed to be secure. Why do these programs require regular updates? [5.4.7]

7. A company has two divisions: Widgets and Gizmos . A new intern has been assigned to the Widget division to examine the Widget design procedure for compliance and talk to her manager about any violations. Later in the summer, she will be assigned to update manuals in the Gizmo division. To do her work, she needs permissions to files. Read permissions means that a user can read files for a given division. Write permissions means that a user can modify or create files for a given division.

   You are the system administrator and need to assign her file permissions today. What permissions (read, write, none, or both) should be assigned to her for each division (Widgets and Gizmos)? [5.4.10]

8. Suppose that a large social network, such as Facebook, made all its data publicly available, but anonymized in the following way: All real-world personal or geographic identifiers, such as real names, phone numbers, social security numbers, birth dates, zip codes, cities, states, and so on, would be modified by replacing them with a specific identifier for each unique category of information. For example, everyone living in London would have their "city" replaced with the same random number (so that the relationships in the network are still intact).

Suppose you were given this data set. Using your real-world resources, is there any legal way you could identify an acquaintance's profile? Why or why not? [5.4.11]

9. What's one reason why the computers of typical home users are not valuable to cyber attackers? [5.4.3]

10. Oh no! Someone just leaked your company's encryption algorithm and now anybody can decrypt your company's secrets! After the breach, you are tasked with re-implementing the encryption system that your company uses to be more resilient to such leaks. How does your new system prevent a similar leak from being so catastrophic in the future? [5.4.12]

11. Why do anti-malware companies like McAfee and Avast push out updates so frequently? [5.4.8]

12. Sally is the network administrator for the network at GlobalDyne, a defense contractor. GlobalDyne's network has top of the line defenses to protect against leaks, including a firewall with deep packet inspection, mandatory access control on all files, security guards, encrypted backups and least privilege access control policies. Nevertheless, multiple high-profile leaks have occurred. How could this be happening? [5.4.13]

13. An intern has been added to your team with the assignment of implementing the encryption module for your application. What advice would you give them regarding the encryption algorithm? [5.4.13]

14. What is the best thing you can do to improve the security of a laptop that must regularly be turned off and stored somewhere where it might get stolen? [5.4.1]

15. Even though it is in space and unconnected to the Internet, the ISS (International Space Station) computers have been infected with malware. How could this happen? [5.4.2]

16. Consider this pseudocode:

```
function  find (dir, name) {
   List = system.run("ls " + dir)
   For file in list {
      If "name" in file {
         Return file
      }
   }
}
```

Assuming that the functions used in the code do not, themselves, have any security flaws, what is the most critical flaw in find()? [5.4.14]

17. A team is developing a new device including its software. What's the best way for them to make sure that the product they ship is as secure as possible? [5.4.7]

18. A network of wireless sensors in an airplane needs to periodically send four unsigned 32 bit values (128 bits total) to a central control computer over a potentially hostile network. When the messages arrive at the computer, there is no source information included in the header, so the message itself must somehow identify the sender.

    Consider the following design proposal: Each sensor has a unique symmetric encryption key, which is hard-coded into the sensor and and the central computer at installation time. Messages from the sensors are encrypted using a strong cipher in the basic ECB (Electronic Code Book) mode, resulting in two 128-bit

blocks of ciphertext (the second block is padding). When messages arrive at the central computer, the computer decrypts new messages with every sensor key until this process results in a valid message. This achieves both decryption and identification of the sender.

What is the worst problem with this design? [5.4.16]

19. You are developing a system that uses user-defined passwords. You've already implemented best practices for password storage and testing (a password-specific hash function and salts) but you'd like the authentication system to be more secure. How could you improve the system? [5.4.17]

20. Assuming your browser has no known vulnerabilities, what is the greatest danger to you, the user, when accessing unencrypted websites? [5.4.1]

References

[1] V. L. Almstrum, P. B. Henderson, V. Harvey, C. Heeren, W. Marion, C. Riedesel, L.-K. Soh, and A. E. Tew. "Concept Inventories in Computer Science for the Topic Discrete Mathematics". In: *SIGCSE Bull.* 38.4 (June 2006), pp. 132–145. ISSN: 0097-8418. DOI: 10.1145/1189136.1189182. URL: http://doi.acm.org/10.1145/1189136.1189182 (cit. on p. 3).

[2] R. Caceffo, S. Wolfman, K. S. Booth, and R. Azevedo. "Developing a Computer Science Concept Inventory for Introductory Programming". In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. SIGCSE '16. Memphis, Tennessee, USA: ACM, 2016, pp. 364–369. ISBN: 978-1-4503-3685-7. DOI: 10.1145/2839509.2844559. URL: http://doi.acm.org/10.1145/2839509.2844559 (cit. on p. 8).

[3] J. W. Carey, M. Morgan, and M. J. Oxtoby. "Intercoder Agreement in Analysis of Responses to Open-Ended Interview Questions: Examples from Tuberculosis Research". In: *CAM Journal* 8.3 (1996), pp. 1–5. DOI: 10.1177/1525822X960080030101. eprint: https://doi.org/10.1177/1525822X960080030101. URL: https://doi.org/10.1177/1525822X960080030101 (cit. on p. 6).

[4] T. Hak and T. Bernts. “Coder training: Theoretical training or practical socialization?” In: *Qualitative Sociology* 19 (June 1996), pp. 235–257. DOI: 10.1007/BF02393420 (cit. on p. 6).

[5] G. L. Herman, M. C. Loui, and C. Zilles. “Creating the Digital Logic Concept Inventory”. In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. SIGCSE ’10. Milwaukee, Wisconsin, USA: ACM, 2010, pp. 102–106. ISBN: 978-1-4503-0006-3. DOI: 10.1145/1734263.1734298. URL: http://doi.acm.org/10.1145/1734263.1734298 (cit. on p. 9).

[6] G. L. Herman, C. Zilles, and M. C. Loui. “A psychometric evaluation of the digital logic concept inventory”. In: *Computer Science Education* 24.4 (2014), pp. 277–303. DOI: 10.1080/08993408.2014.970781. eprint: https://doi.org/10.1080/08993408.2014.970781. URL: https://doi.org/10.1080/08993408.2014.970781 (cit. on p. 9).

[7] D. Hestenes, M. Wells, and G. Swackhamer. “Force concept inventory”. In: *The Physics Teacher* 30.3 (1992), pp. 141–158. DOI: 10.1119/1.2343497. eprint: https://doi.org/10.1119/1.2343497. URL: https://doi.org/10.1119/1.2343497 (cit. on pp. 2, 3).

[8] I. Ion, R. Reeder, and S. Consolvo. ““...No one Can Hack My Mind”: Comparing Expert and Non-Expert Security Practices”. In: *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. Ottawa: USENIX Association, 2015, pp. 327–346. ISBN: 978-1-931971-249. URL: https://www.usenix.org/conference/soups2015/proceedings/presentation/ion (cit. on p. 11).

[9] J. Jang-Jaccard and S. Nepal. “A survey of emerging threats in cybersecurity”. In: *Journal of Computer and System Sciences* 80.5 (2014). Special Issue on Dependable and Secure Computing, pp. 973–993. ISSN: 0022-0000. DOI:

https://doi.org/10.1016/j.jcss.2014.02.005. URL: http://www.sciencedirect.com/science/article/pii/S0022000014000178 (cit. on p. 23).

[10] K. KELLEY, B. CLARK, V. BROWN, and J. SITZIA. "Good practice in the conduct and reporting of survey research". In: *International Journal for Quality in Health Care* 15.3 (May 2003), pp. 261–266. ISSN: 1353-4505. DOI: 10.1093/intqhc/mzg031. eprint: http://oup.prod.sis.lan/intqhc/article-pdf/15/3/261/5251095/mzg031.pdf. URL: https://doi.org/10.1093/intqhc/mzg031 (cit. on p. 4).

[11] K. Malterud, V. Siersma, and A. D. Guassora. "Sample Size in Qualitative Interview Studies: Guided by Information Power." In: *Qualitative health research* (2015) (cit. on pp. 5, 15).

[12] A. Narayanan and V. Shmatikov. "Robust De-anonymization of Large Sparse Datasets". In: *Proceedings of the 2008 IEEE Symposium on Security and Privacy*. SP '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 111–125. ISBN: 978-0-7695-3168-7. DOI: 10.1109/SP.2008.33. URL: https://doi.org/10.1109/SP.2008.33 (cit. on p. 27).

[13] R. Popping. "Analyzing Open-ended Questions by Means of Text Analysis Procedures". In: *Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique* 128.1 (2015), pp. 23–39. DOI: 10.1177/0759106315597389. eprint: https://doi.org/10.1177/0759106315597389. URL: https://doi.org/10.1177/0759106315597389 (cit. on pp. 5, 6, 17).

[14] R. Popping. "Some views on agreement to be used in content analysis studies". In: *Quality & Quantity* 44.6 (Oct. 2010), pp. 1067–1078. ISSN: 1573-7845. DOI:

10.1007/s11135-009-9258-3. URL: https://doi.org/10.1007/s11135-009-9258-3 (cit. on p. 6).

[15] R. Reeder, I. Ion, and S. Consolvo. "152 Simple Steps to Stay Safe Online: Security Advice for Non-tech-savvy Users". In: *IEEE Security Privacy* (2018), pp. 1–1. ISSN: 1540-7993. DOI: 10.1109/MSP.2017.265093101 (cit. on p. 11).

[16] A. T. Sherman, L. Oliva, D. DeLatte, E. Golaszewski, M. Neary, K. Patsourakos, D. S. Phatak, T. Scheponik, G. L. Herman, and J. Thompson. "Creating a Cybersecurity Concept Inventory: A Status Report on the CATS Project". In: *CoRR* abs/1706.05092 (2017). arXiv: 1706.05092. URL: http://arxiv.org/abs/1706.05092 (cit. on p. 7).

[17] K. C. Webb and C. Taylor. "Developing a Pre- and Post-course Concept Inventory to Gauge Operating Systems Learning". In: *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. SIGCSE '14. Atlanta, Georgia, USA: ACM, 2014, pp. 103–108. ISBN: 978-1-4503-2605-6. DOI: 10.1145/2538862.2538886. URL: http://doi.acm.org/10.1145/2538862.2538886 (cit. on p. 9).

[18] Wikipedia. *Outline of Computer Security*. 2019. URL: https://en.wikipedia.org/wiki/Outline_of_computer_security (visited on 05/05/2019) (cit. on p. 14).