



May 20th, 3:00 PM

Measuring Hacking Ability Using a Conceptual Expertise Task

Justin S. Giboney

School of Business, University at Albany, jgiboney@albany.edu

Jeffrey G. Proudfoot

Information and Process Management, Bentley University, jproudfoot@bentley.edu


Sanjay Goel

School of Business, University at Albany, goel@albany.edu

Joseph S. Valacich

Eller College of Management, The University of Arizona, jsvalacich@cmi.arizona.edu

Follow this and additional works at: <https://commons.erau.edu/adfsl>

 Part of the [Aviation Safety and Security Commons](#), [Computer Law Commons](#), [Defense and Security Studies Commons](#), [Forensic Science and Technology Commons](#), [Information Security Commons](#), [National Security Law Commons](#), [OS and Networks Commons](#), [Other Computer Sciences Commons](#), and the [Social Control, Law, Crime, and Deviance Commons](#)

Scholarly Commons Citation

Giboney, Justin S.; Proudfoot, Jeffrey G.; Goel, Sanjay; and Valacich, Joseph S., "Measuring Hacking Ability Using a Conceptual Expertise Task" (2015). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 8.

<https://commons.erau.edu/adfsl/2015/wednesday/8>

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

EMBRY-RIDDLE
Aeronautical University™
SCHOLARLY COMMONS

(c)ADFSL



MEASURING HACKING ABILITY USING A CONCEPTUAL EXPERTISE TASK

Justin Scott Giboney
School of Business
University at Albany
Albany, NY 12222
jgiboney@albany.edu

Sanjay Goel
School of Business
University at Albany
Albany, NY 12222
goel@albany.edu

Jeffrey Gainer Proudfoot
Information and Process Management
Bentley University
Waltham, MA 02452
jproudfoot@bentley.edu

Joseph S. Valacich
Eller College of Management
The University of Arizona
Tucson, AZ 85721
jvalacich@cmi.arizona.edu

ABSTRACT

Hackers pose a continuous and unrelenting threat to organizations. Industry and academic researchers alike can benefit from a greater understanding of how hackers engage in criminal behavior. A limiting factor of hacker research is the inability to verify that self-proclaimed hackers participating in research actually possess their purported knowledge and skills. This paper presents current work in developing and validating a conceptual-expertise based tool that can be used to discriminate between novice and expert hackers. The implications of this work are promising since behavioral information systems researchers operating in the information security space will directly benefit from the validation of this tool.

Keywords: hacker ability, conceptual expertise, skill measurement

1. INTRODUCTION

Governments, businesses, universities and other organizations are prime targets for hackers. A common motive for hackers to target these organizations is data theft [1] which results in billions of dollars of losses annually [2]. Due to the threat that hackers pose to organizations, researchers have been encouraged to investigate hacker motives and behavior [3]. There have been recent attempts to further understand hacker behavior e.g., [2], [4], [5]. However, these studies rely on data collected from *self-reported* hackers. Respondents can pose as hackers due to gain the incentives provided during data collection. It is unverifiable whether the samples utilized in prior research are based on data collected from *actual* hackers or whether these samples are based on data collected from persons misrepresenting their

hacking abilities and experience. A consequence of this uncertainty is the questionable validity and generalizability of the findings reported in prior hacking research.

A second issue is the tendency for researchers to lump all hackers into a single category during data analysis. This is typically done as a means of comparing hackers to other groups, but previous research indicates that there is more than one type of hacker [6]. Categories of hackers include: script kiddies, petty thieves, virus writers, professional criminals, and government agents [6], [7]. Furthermore, the motivations and skill levels of different types of hackers are varied [6]. In light of these differences, hacking researchers would benefit tremendously from the ability to more accurately measure each hacker's level of skill. The ability to measure hacking skill would allow

researchers to verify that a self-proclaimed hacker indeed possesses requisite technical skills. It would also allow analyses to be conducted on subsets of data for different groups of hackers based on their level of skill and areas of expertise.

In short, there is currently no scientific measure that can be used to assess hacking skill level without employing qualitative research methods (e.g., interviews) [8]. While effective, qualitative methods are much less scalable than survey-based methods as surveys can be administered widely with few temporal or geographic limitations. Furthermore, hacking activities often require behavior that is criminal in nature; a survey-based methodology for data collection may elicit a more candid response from a participant since the identity of the respondent can remain anonymous.

The goal of this research is to develop a survey-based methodology for determining a hacker's skill level using an 18-scenario scale. If a scale can be developed to measure a hacker's skill level, researchers can (1) more accurately discriminate between categories of hackers, (2) more accurately quantify who is a hacker and who is not, and (3) provide evidence that their findings are indeed generalizable to the population of interest.

The scale development process used for this research is in accordance with recognized scale development protocols [9] and is based on measuring conceptual expertise, an approach previously utilized by researchers in a variety of disciplines [10], [11]. Upon completing scale development, this research proposes to collect and analyze data to validate the accuracy of the measurement tool. This paper presents an overview of the scale development process concerning the validity of this novel approach to measuring hacker ability.

2. SCALE DEVELOPMENT

There are six recommended phases used for scale development: 1) conceptualization, 2) development of measures, 3) model specification, 4) scale evaluation and refinement, 5) validation, and 6) norm development [9]. Each of these steps is discussed in the following sections.

2.1 Conceptualization

The goal of the conceptualization phase is to provide a precise definition of the construct of interest and establish conceptual arguments for how the construct can be discriminated from previously-specified and evaluated constructs found in literature [9]. This paper introduces hacking conceptual expertise as a new construct based on the conceptual expertise construct found in cognitive science literature [10], [12]. Hacking conceptual expertise is comparable to, and should distinguish from, two similar constructs: computer self-efficacy [13] and computer ability [14]. This section will first discuss expertise before addressing computer self-efficacy and computer ability.

Expertise is a “manifestation of skills and understanding resulting from the accumulation of a large body of knowledge” [12, p. 167]. A hacker's expertise is manifested in their ability to write code or scripts that can circumvent security protocols, disrupt the intended functions of a system, collect valuable information, and not get caught [6]. Many hackers are novices, sometimes referred to as “script kiddies”, who have only a surface understanding of hacking but still employ software and scripts written by experts to perform their attacks [6], [15]. Expert hackers understand hacking at a deeper level as they have a command of the common weaknesses and vulnerabilities of information systems. Therefore, we formally define the construct *hacking conceptual expertise* as the manifestation of skills and understanding about circumventing security protocols, disrupting the intended functions of systems, collecting valuable information, and not getting caught.

Computer self-efficacy is a similar construct to hacking conceptual expertise and is based on Social Cognitive Theory (SCT). SCT explains that social pressures, context, cognitive and personal characteristics, and behavior are reciprocally determined. [13], [16]. Self-efficacy is part of the cognitive and personal characteristics that drive behavior [13], [17]. Self-efficacy is a belief people have about their capacity to perform an action and their skill at performing the action [13], [17]. According to SCT, people are more likely to act if they believe that there will be positive outcomes as a result of their action [13], [17]. If people believe that they are good at an action, they are more likely to believe that they

will receive a positive outcome by performing the action, and therefore, will be more likely to perform the action [13], [17]. Computer self-efficacy is the belief people have about their capacity to perform actions that accomplish a computer-based task [13]. Computer self-efficacy has three main components: magnitude, strength, and generalizability [13]. Magnitude refers to the perception that people can accomplish more difficult tasks [13]. Strength refers to the confidence people have in being able to perform the tasks [13]. Generalizability refers to the extent to which a person's judgments include multiple activities [13].

Another similar measure to hacking conceptual expertise is computer ability [14]. Computer ability is based on two concepts: how important a skill is to a task and the perceived skill level of the individual in performing a task [14]. This research will demonstrate that hacking conceptual expertise is both distinct from and an antecedent to computer self-efficacy and computer ability.

2.2 Development of Measures

Measurement development is a process traditionally completed in two steps. First, a set of items that represent the construct is generated, and second, the content validity of the set of items is assessed [9]. Before discussing the generation of items, this paper will first discuss how these items will be used in a specialized task to measure hacker conceptual expertise rather than using a traditional survey.

2.2.1 The Conceptual Expertise Task

When people solve problems they approach tasks based on the mental representation they have of the problem [10], [18]. Their mental representations are based on stored information (memories) that assist in knowledge-based decisions [19]. People use stored information as nodes (or waypoints) that allow them to follow a solution path [20], [21]. Therefore, problem solvers use the mental representations they have of the task to find a solution. For example, chess masters can identify more ways to achieve checkmate (where the nodes are necessary moves) than novice chess players.

Experts perform tasks better than novices as they have superior mental representations of problems; this is attributable to larger quantities of stored information and solution nodes (i.e., steps to achieve the solution) [10]. In other words, experts have more strategies at their disposal to find solutions to problems as compared with novices. Due to larger quantities of stored information, experts typically organize information into abstract categories [10]. Abstract categories are used to filter tasks and potential solutions; they allow experts to more quickly and efficiently solve problems [10]. These abstract categories form the basis for testing expertise.

The abstract categorization of experts can be leveraged to distinguish between experts and novices. When given a set of related problems (e.g., end of chapter questions in a textbook) and asked to group the problems, experts will rely on their abstract categories to organize the problems based on principles of the domain [12]. When novices are given the same task, however, they will organize the problems based on physical evidence, explicit words, or formulas [12]. For example, in a study distinguishing between expert and novice programmers, experts sorted programming problems by solution algorithms and novices sorted the same problems by application areas [22].

Researchers can capitalize on this difference between experts and novices (abstract categories versus physical evidence) to create a scoring system to measure expertise [10], [11]. To create this scoring system, underlying principles from the domain of expertise are derived from literature, textbooks, and other related sources. These underlying principles are termed *deep features* as they show understanding of a given domain (e.g., social engineering). Deep features are contrasted with *surface features* that are the objects or contexts (e.g., stealing financial data) represented in a problem [10]. When participants group deep features together more often than they group surface features, the participants are considered experts. When participants group surface features together more often than they group deep features, the participants are considered novices.

The conceptual knowledge task is typically performed using a card sort of relevant scenarios

on 3x5 cards with each card having one deep feature and one surface feature [11]. An example of a scenario with a deep feature of system resource consumption and a surface feature of financial data is as follows:

Eve sends out requests to millions of machines using an IP address assigned to a server at a stock brokerage.

While this is an example of a scenario that could be displayed on one card, assume that a researcher creates 18 cards lettered A-R, each possessing a unique hacking-related scenario containing a deep feature and a surface feature. The hypothesized groupings could look something like Table 1. Participants, without seeing Table 1 or knowing the hypothesized features, are asked to sort the cards into groups with the following restrictions:

- You must create more than one group
- Each group must have at least 2 cards and fewer than 15 cards
- Each card can only be a part of 1 group

- Create a name for your groups

Table 2 provides an example of how a participant might group the scenarios. Once grouped, researchers can score the pairings of every combination in each group to classify it as a surface feature pair (S), a deep feature pair (D), or an unexpected pair (U). For example, the pair P-G in the participant’s first group is a surface feature pair as both scenarios are in the “usernames and passwords” column. The pair L-I in the participant’s second group is a deep feature pair as both scenarios are in the “input validation” row. The pair F-E in the participant’s third group is an unexpected pair as the two scenarios are neither in the same row nor the same column. In total, this participant identified 6 deep pairings, 12 surface pairings, and 18 unexpected pairings. This participant is likely more novice than expert as he or she identified more surface features than deep features. However, the participant could have created an even higher number of surface feature pairs, thus he or she is likely not a complete novice.

Table 1 Example problem matrix

		Hypothesized surface features		
		Fake website	Usernames and passwords	Financial data
Hypothesized deep features	Authentication/ Authorization	H	D	O
	Hiding tracks	F	N	A
	Input validation/ Memory override	Q	J	E
	Resource consumption	M	P	R
	Social engineering	K	G	C
	Vulnerability detection	B	L	I

2.2.2 Generating items for the hacking conceptual expertise task

The next step in scale development is to “generate a set of items that fully represent the conceptual

domain of the construct” [9, p. 304]. For the conceptual expertise task, the generation of items begins with the identification of deep features. Previous researchers using the conceptual expertise task have used textbook problems to

identify deep features see [10], [11]. As information security textbooks do not typically provide information on how to engage in criminal hacking behavior, we relied on both information security textbooks and academic literature addressing hacking scenarios to generate deep

features for the hacking conceptual exercise. Table 3 contains a thorough, but not exhaustive, list of vulnerabilities and security measures identified in the set of textbooks and relevant literature used for this study [1], [4], [6], [8], [23]–[40].

Table 2 Example participant grouping result

Group 1 – Hacks that involve numerous targets P, O, M, J, Q, G, D
Group 2 – Hacks that involve hacker input N, L, A, I
Group 3 – Hacks that involve pretending to be someone else or pretending to do something good F, E, B, R
Group 4 – Hacks that involve programming K, H, C

Table 3 Hacks, vulnerabilities, and security measures referenced in relevant literature

Authentication/Authorization	Encryption, Security tokens, Permissions, Password cracking, Two-step commit, Certificate authorities, Password salting, Keystroke logging, Rainbow tables, Brute force attacks
Hiding tracks	Malware signatures, Removing log files, Audit-disabling software, Disabling security controls, Using proxies, IP spoofing, Steganography
Input validation/Memory override	Buffer overflow, Cross-site scripting, Maladvertising, SQL injection, Heap spraying, Format string attacks, Dangling pointers
Resource consumption	Denial of service attacks, Syn flood, ACK storm, Email bombs, HTTP POST DDOS, Smurf attacks, Spamming
Social engineering	Spear Phishing, Pharming, Nigerian scam, Phishing
Vulnerability detection	Man in the middle attacks, Port scanning, Ping sweeps, Packet sniffing, Network mapping, War driving, Bluesnarfing
Actions/Outcomes	Electronic espionage, Zombie networks, Spyware, Website defacement, Computer worms, Trojan horses, Root kits, Ransomware, Leak of information, Bot net, Trap doors, Logic bombs

A careful review of the hacks, vulnerabilities, and security measures identified in relevant literature allowed us to organize seven principles of hacking that form the basis for our deep features. In the next section we will empirically test and validate the categorization of these hacking principles. It is worth noting that the last category titled

“Actions/Outcomes” in Table 3 does not contain hacking techniques, but rather contains outcomes of hacking activities. This category was not considered ideal for evaluating a person’s ability to carry out a hacking attempt, but rather how well someone knows about hacking activities in general, therefore, it was excluded from our final

set of deep features. For the conceptual expertise task, deep features are coupled with their corresponding surface features to create a matrix. We created three areas of surface features, namely financial data, fake websites, and usernames/passwords, to correspond with six deep features. Table 4 contains the 18 scenarios resulting from the use of the features contained in the matrix. Recall that Table 1 contains the matrix depicting how the deep features are crossed with the surface features.

2.2.3 Assessing Content Validity

Before using this task to discriminate between novice and expert hackers, the items must first be scientifically validated. The validation process will

be completed using two approaches. First, expert information security practitioners and academics will review our proposed methodology and provide feedback. Second, the scenarios presented in Table 4 will be empirically validated using an item-ranking task. We have already compiled feedback on the hacking conceptual expertise task proposed in this paper; feedback was solicited from four security experts with either an industry or academic background. The general consensus of the polled experts is that this is a feasible approach for discriminating between expert and novice hackers. A common concern is that our approach may only measure how well a hacker conceptually understands hacking methods without directly assessing a hacker's actual ability.

Table 4 Hacking conceptual expertise scenarios

Hack	#	Scenario
Removing log files	A	Eve deletes log files as she combs through a compromised machine looking for tax returns.
Port scanning	B	Eve uses a malicious website to scan for open ports of visitors.
Phishing	C	Eve, pretending to be a bank website, emails Kelly asking for her bank information.
Rainbow tables	D	Eve uses a rainbow table to decrypt secret military intranet links.
SQL injection	E	Eve uses a semicolon in a web form to access user account balances in the database.
Using proxies	F	Eve uses a proxy while creating a website to create a zombie network
Nigerian scam	G	Eve sends Twitter messages en masse asking people to click on an Internet link in return for some secret information.
Certificate authority	H	Eve becomes her own certificate authority as she creates a fictitious e-commerce business.
Man-in-the-middle attack	I	Eve captures Wi-Fi network traffic from a conference to watch for financial transactions.
Improper file validation	J	Eve uploads an executable to a server expecting an image, the executable sends out instant messages with Internet links to random email addresses.
Pharming	K	Eve creates a website similar to a well-known company using a similar domain name.
Ping sweep	L	Eve sends a ping to networked machines and then sends an Internet link as a message to live machines.
HTTP POST DOS attack	M	Eve creates fake websites that post to a targeted website normally, but that are extremely slow (e.g. 1 byte/110 seconds).
Malware signature avoidance	N	Eve has created a virus to look for Internet links to sensitive data stored on a computer that changes itself after every install.
Password salting	O	Eve is attempting to figure out the salt that was used for some financial transactions.
Email DOS	P	Eve created a script to send hundreds of emails with an Internet link using

		fake email addresses to a particular company leader.
Cross-site scripting	Q	Eve posts a response on a forum that allows Eve to redirect users to a malicious website.
Smurf attack	R	Eve sends out requests to millions of machines using the IP address of the server of a stock trading institution.

More specifically, one of the security experts stated that deep features "...are more clear cut than the surface features." Another security expert suggested that the deep pairings may be too intuitive. However, we do not consider these responses to be troubling as experts should consider deep features to be both clear and intuitive. We take these comments as a sign that the measurement method is well specified.

There were a number of other requests from the security experts that we will incorporate in the next iteration of the measurement tool. For example, one security expert suggested that we include script kiddie scenarios that reference the use of existing prepackaged tools. Another security expert suggested that we include more hardware exploits.

Upon incorporating this feedback into a new set of items, we will empirically validate the items by selecting 20 new security experts from a state information security team. To empirically validate the items, each item should be adequately representative of the deep feature to which it is assigned [9], [41], [42]. Mackenzie et al. [9] recommended a technique suggested by Hinkin and Tracey [43] in which a matrix is created with the items in the first column and the deep features listed as column headers. The matrix is then distributed to raters who are asked to rate how well each item fits with each column header on a five-point Likert-type scale (1 = not at all; 5 = completely). Table 5 contains a hypothetical example of the matrix that we will use. Data collection for both of these approaches is currently underway. Preliminary findings from both approaches will be reported at the conference.

Table 5 Hypothetical example of item rating task

Rater # = 001	Authentication /Authorization	Hiding tracks	Input validation /Memory override	Resource consumption	Social engineering	Vulnerability detection
Eve deletes log files as she combs through a compromised machine looking for tax returns.	1	5	1	2	1	2
Eve uses a malicious website to scan for open ports of visitors.	1	1	1	2	2	4
...
Eve sends out requests to millions of machines using the IP address of	2	1	1	4	1	2

the server of a stock trading institution.						
--	--	--	--	--	--	--

2.3 Model Specification

The next phase in scale development is to specify how the indicators capture the expected relationships with the construct [9]. While this stage typically involves specifying formative or reflective indicators for a construct, the conceptual expertise task does not treat the indicators as formative measures in a scale, rather they are used to calculate a single expertise score. Therefore, our model specification will be the percentage of deep pairs identified by a participant compared to the percentage of surface pairs identified by the participant.

2.4 Scale Evaluation and Refinement

Scale evaluation and refinement is a two-step process based on (1) conducting a pilot study and (2) modifying items in the survey. After revising the scenarios from the feedback we receive from experts participating in our item-validation tasks, we will conduct a pilot study comprised of 20 security experts, 20 novices, and 20 claimed hackers on Amazon’s Mechanical Turk. Security experts will be selected from government and corporate information security teams with connections to the university. Novices will be selected from introductory Computer Science, Informatics, or Information Systems courses. The pilot study will allow us to further refine the scale by adjusting the scenarios based on our results. We will look to refine scenarios that are paired by experts using surface features as well as scenarios paired by novices based on deep features. We will also look for scenarios commonly paired in unexpected ways. The item-refinement process is iterative and will be carried out until the scale possesses sufficient discriminatory power.

2.5 Validation

Validation is a three-step process comprised of the following tasks: 1) gathering data from a complete sample, 2) assessing scale validity, and 3) cross-validating the scale [9]. As the scenarios will

evolve throughout the pilot-testing process, we will conduct the main data collection with a full-sized sample. We will sample 50 security experts, 50 novices, and 50 self-identified hackers from Amazon’s Mechanical Turk.

Once the full-sized sample is collected we can assess the scale’s validity in two ways. First, we will be able to use a known-group comparison method, and second, we will assess the nomological validity of the scale. The known-group comparison is the use of groups (novices and experts) that should demonstrate differences on the scale [9]. We expect that novices will create more surface pairings than experts, and that novices will create less deep pairings than experts.

To assess nomological validity we will measure how well hacking conceptual expertise relates to similar measures. Specifically, we expect hacking conceptual expertise to increase perceptions of computing ability measured through computer self-efficacy and computer ability (see Figure 1).

2.6 Norm Development

The last step in scale development is to develop norms for the scale. This involves discovering the distribution of the scores from different populations. While we currently do not have plans to create norms for this scale, we are optimistic that this paper will serve as a foundation for developing norms in future work.

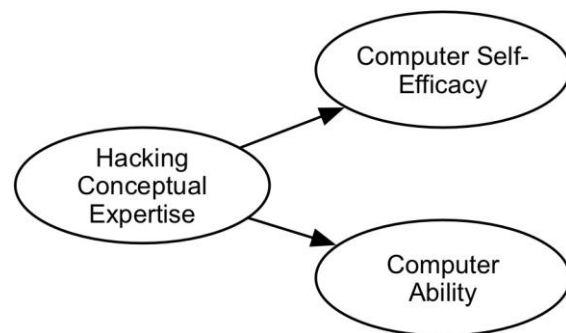


Figure 1 Measurement model

3. CONCLUSION

Hackers continue to pose a serious threat to organizations. Security researchers can benefit from a greater understanding of how and why hackers engage in criminal behavior. A limiting factor of such studies is the inability to verify that self-proclaimed hackers participating in research actually possess their purported knowledge and skills. This paper presents a cogent plan to develop and validate a conceptual-expertise based tool that can be used to discriminate between novice and expert hackers.

The proposed tool operates on the premise that given a set of scenarios, experts will rely on their understanding of abstract categories to organize problems based on principles of the domain whereas novices organize problems based on physical evidence, explicit words, or formulas. In other words, experts will group items based on deeper features while novices will group items based on surface features. To create a conceptual-expertise based tool for measuring hacker ability that possesses sufficient discriminatory power, items must first be developed and validated. We have developed 18 scenarios and are in the process of refining both the task and the scenarios by soliciting feedback from information security experts. These 18 scenarios will be a scale that can be used in survey-based research to measure hacker skill level. Once feedback from solicited experts is analyzed, our model will be refined followed iterative pilot testing and data collection.

The implications of this work are promising as behavioral information systems researchers operating in the information security space will directly benefit from the validation of this tool. Furthermore, adaptations of this tool have the potential to be utilized in a variety of contexts and applications in information systems research.

REFERENCES

- [1] Verizon, "2012 data breach investigations report," 2012.
- [2] D. Dey, A. Lahiri, and G. Zhang, "Hacker behavior, network effects, and the security software market," *J. Manag. Inf. Syst.*, vol. 29, no. 2, pp. 77–108, Oct. 2012.
- [3] M. A. Mahmood, M. Siponen, D. Straub, H. R. Rao, and T. S. Raghu, "Moving toward black hat research in Information Systems security: An editorial introduction to the special issue," *MIS Q.*, vol. 34, no. 3, pp. 431–433, 2010.
- [4] Z. Xu, Q. Hu, and C. Zhang, "Why computer talents become computer hackers," *Commun. ACM*, vol. 56, no. 4, p. 64, Apr. 2013.
- [5] J. S. Giboney, A. Durcikova, and R. W. Zmud, "What motivates hackers? Insights from the Awareness-Motivation-Capability Framework and the General Theory of Crime," in *Dewald Roode Information Security Research Workshop*, 2013, pp. 1–40.
- [6] M. K. Rogers, "A two-dimensional circumplex approach to the development of a hacker taxonomy," *Digit. Investig.*, vol. 3, no. 2, pp. 97–102, Jun. 2006.
- [7] R. Chiesa and S. Ducci, *Profiling Hackers: The Science of Criminal Profiling as Applied to the World of Hacking*. Boca Raton, FL: Auerbach Publications, 2009.
- [8] A. E. Voiskounsky and O. V Smyslova, "Flow-based model of computer hackers' motivation.," *CyberPsychology Behav.*, vol. 6, no. 2, pp. 171–180, Apr. 2003.
- [9] S. B. Mackenzie, P. M. Podsakoff, and N. P. Podsakoff, "Construct measurement and validation procedures in MIS and behavioral research: Integrating new and existing techniques," *MIS Q.*, vol. 35, no. 2, pp. 293–334, 2011.
- [10] M. T. H. Chi and P. J. Feltovich, "Categorization and representation of physics problems by experts and novices," *Cogn. Sci.*, vol. 5, no. 2, pp. 121–152, 1981.
- [11] J. I. Smith, E. D. Combs, P. H. Nagami, V. M. Alto, H. G. Goh, M. A. A. Gourdet, C. M. Hough, A. E. Nickell, A. G. Peer, J. D. Coley, and K. D. Tanner, "Development of the Biology Card Sorting Task to Measure Conceptual Expertise in Biology," *CBE-Life Sci. Educ.*, vol. 12, no. 4, pp. 628–644, Dec. 2013.
- [12] M. T. H. Chi, "Laboratory methods for assessing experts' and novices' knowledge," in *The Cambridge Handbook of Expertise and Expert Performance*, K. A. Ericsson, N. Charness, P. J. Feltovich, and R. R. Hoffman, Eds. Cambridge University Press, 2006, pp. 167–184.

- [13] D. R. Compeau and C. A. Higgins, "Computer self-efficacy: Development of a measure and initial test," *MIS Q.*, vol. 19, no. 2, pp. 189–211, 1995.
- [14] P. H. Cheney and R. R. Nelson, "A tool for measuring and analyzing end user computing abilities," *Inf. Process. Manag.*, vol. 24, no. 2, pp. 199–203, 1988.
- [15] T. J. Holt, "Subcultural evolution? Examining the influence of on- and off-line experiences on deviant subcultures," *Deviant Behav.*, vol. 28, no. 2, pp. 171–198, Feb. 2007.
- [16] A. Bandura, "Social cognitive theory: An agentic perspective," *Annu. Rev. Psychol.*, vol. 52, no. 1, pp. 1–26, 2001.
- [17] A. Bandura, "Self-efficacy mechanism in human agency," *Am. Psychol.*, vol. 37, no. 2, pp. 122–147, 1982.
- [18] A. Newell and H. A. Simon, *Human Problem Solving*. Englewood Cliffs, NJ: Prentice Hall, 1972.
- [19] A. Chandra and R. Krovi, "Representational congruence and information retrieval: Towards an extended model of cognitive fit," *Decis. Support Syst.*, vol. 25, pp. 271–288, 1999.
- [20] H. A. Simon and J. R. Hayes, "Understanding written problem instructions.," in *Knowledge and Cognition*, L. W. Gregg, Ed. Potomac, MD: Lawrence Erlbaum Associates, 1974, pp. 165–200.
- [21] J. R. Hayes and H. A. Simon, "Psychological differences among problem isomorphs," in *Cognitive Theory*, 2nd ed., J. N. Castellan Jr., D. B. Pisoni, and G. R. Potts, Eds. Hillsdale, NJ: Lawrence Erlbaum Associates, 1977, pp. 21–41.
- [22] M. Weiser and J. Shertz, "Programming problem representation in novice and expert programmers," *Int. J. Man. Mach. Stud.*, vol. 19, no. 4, pp. 391–398, 1983.
- [23] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2006.
- [24] M. T. Goodrich and R. Tamassia, *Introduction to Computer Security*. Boston, MA: Pearson Education, Inc., 2011.
- [25] T. Jordan, "Mapping Hacktivism: Mass Virtual Direct Action (MVDA), Individual Virtual Direct Action (IVDA) And Cyberwars," *Comput. Fraud Secur.*, vol. 4, no. 1, pp. 8–11, 2001.
- [26] T. Jordan and P. Taylor, "A sociology of hackers," *Sociol. Rev.*, vol. 46, no. 4, pp. 757–780, Nov. 1998.
- [27] S. M. Furnell and M. J. Warren, "Computer hacking and cyber terrorism: The real threats in the new millennium?," *Comput. Secur.*, vol. 18, no. 1, pp. 28–34, 1999.
- [28] O. Turgeman-Goldschmidt, "Hackers' Accounts: Hacking as a Social Entertainment," *Soc. Sci. Comput. Rev.*, vol. 23, no. 1, pp. 8–23, Feb. 2005.
- [29] V. Mookerjee, R. Mookerjee, A. Bensoussan, and W. T. Yue, "When hackers talk: Managing information security under variable attack rates and knowledge dissemination," *Inf. Syst. Res.*, vol. 22, no. 3, pp. 606–623, 2011.
- [30] D. P. Twitchell, "Augmenting detection of social engineering attacks using deception detection technology," in *International Conference on i-Warfare and Security*, 2006.
- [31] R. E. Bell, "The prosecution of computer crime," *J. Financ. Crime*, vol. 9, no. 4, pp. 308–325, 2002.
- [32] H. Liang and Y. Xue, "Avoidance of information technology threats: A theoretical perspective," *MIS Q.*, vol. 33, no. 1, pp. 71–90, 2009.
- [33] G. B. Magklaras and S. M. Furnell, "Insider threat prediction tool: Evaluating the probability of IT misuse," *Comput. Secur.*, vol. 21, no. 1, pp. 62–73, 2002.
- [34] Symantec Corporation, "Internet Security Threat Report," Mountain View, California, 2013.
- [35] L. Holmlund, D. Mucisko, K. Kimberland, and J. Freyre, "2010 cybersecurity watch survey: Cybercrime increasing faster than some company defenses," 2010.
- [36] CyberEdge Group, "2014 Cyberthreat Defense Report," 2014.
- [37] R. T. Wright and K. Marett, "The influence of experiential and dispositional factors in phishing: An empirical investigation of the deceived," *J. Manag. Inf. Syst.*, vol. 27, no. 1, pp. 273–303, 2010.
- [38] B. Parmar, "Protecting against spear-phishing," *Comput. Fraud Secur.*, vol. 2012, no. 1, pp. 8–11, Jan. 2012.

- [39] S. Goel and H. A. Shawky, “Estimating the market impact of security breach announcements on firm values,” *Inf. Manag.*, vol. 46, no. 7, pp. 404–410, Oct. 2009.
- [40] R. Boyle and J. G. Proudfoot, *Applied Information Security: A Hands-On Guide to Information Security Software*, 2nd ed. New Jersey: Pearson, 2014.
- [41] F. N. Kerlinger and H. B. Lee, *Foundations of Behavioral Research*, 4th ed. New York: Cengage Learning, 1999.
- [42] D. W. Straub, M.-C. Boudreau, and D. Gefen, “Validation guidelines for IS positivist research,” *Commun. Assoc. Inf. Syst.*, vol. 13, no. 1, pp. 380–427, 2004.
- [43] T. R. Hinkin and J. B. Tracey, “An analysis of variance approach to content validation,” *Organ. Res. Methods*, vol. 2, no. 2, pp. 175–186, 1999.

