Cybersecurity: Human-Computer Interaction (HCI) Theories and Issues in Design

Prepared by:

Uyen Bui

Luis Cruz Martir

Carmen Machado

Dawson Ohligschlager

California State University, Long Beach

Department of Psychology

# **Table of Contents**

[**Executive Summary**](#_yvyl1wxdgg0m)3

[**Theories**](#_j29k9sq2eziz)4

[Stages-of-Action Theory](#_y88kjouze39x) 4

[Dynamic Theory](#_f47l28houi9) 6

[**Key Issues in Design**](#_fs5kfexufgyq)7

[Complexity](#_my3stt5k6wz1) 7

[Authentication](#_2invk7n8zu5g) 8

[Updates](#_vg6bcn9pg1to) 9

[**Practical Applications**](#_q0l5lp3hxlrp)10

[Google Authenticator - Complexity](#_6ca7eyz5s29u) 10

[Google Authenticator - Authentication](#_hcuxfb65nni4) 11

[Google Authenticator - Updates](#_88qjprb85o92) 11

[Areas of Improvement and Recommendations - Google Authenticator](#_52wp0dkd6aum) 12

[Recommendations - Artificial Pancreas](#_n3wljf16oq17) 13

[Recommendations - Hawaii Emergency Management Agency Interface](#_1lpxxmy1o65p) 15

[**Appendix A - Recorded Consultation Session Access**](#_bvtyvk40f282)17

[**References**](#_zf9lxncffe75)18

# **Executive Summary**

Cybersecurity remains important, given the amount of information that can be accessed without proper authorization. The purpose of this current report is to review relevant human-computer interaction (HCI) theories to the field of cybersecurity, identify key issues and apply them to one cybersecurity interface, and identify areas of improvement for that interface. Recommendations made for each outstanding issue were applied to two additional cybersecurity interfaces. Key findings of the research review are:

* Focus should be placed on the latter steps in the stages-of-action theory, where adequate feedback should be provided to the user. Dynamic theory is important to help understand who has used the system in an applied setting, how the attributes are being defined by the user, and to foster adoption of new products with time and conscientiousness.
* Task complexity, authentication, and updates were found to be most impactful issues in the design of cybersecurity systems.
* Google Authenticator was evaluated based on the identified issues and we recommended that visibility of system status, adequate feedback, error prevention, and clear help and documentation be incorporated into the design.
* Artificial Pancreas interface recommendations to help prevent Do-it-Yourself (DIY) Hacking are, (1) continuous access; (2) information confidentiality; (3) integrity without data alteration; (4) privileged access identification; and (5) identity verification before executing a task.
* Hawaii Emergency Management Agency (HI-EMA) interface recommendations include communicating which type of alert will be sent out or was sent out based on feedback, clear distinguishment between real alert and test alerts, and providing help for when a false alert is sent out.

# **Theories**

## Stages-of-Action Theory

The stages-of-action theory of HCI states that specific actions users go through as they interact with user interfaces (UI) or products should be considered when designing said interfaces and/or products (Shneiderman et al., 2016). An example of a general set of actions is provided by Norman (2013): (1) Form a goal; (2) Identify a sequence of actions; (3) Carry out the action; (4) Perceive the system state; (5) Interpret the system state; and (6) Evaluate the outcome of the system state with regards to goals and intentions.

With this general outline of actions, designers can describe how users will interact with an interface (Polson & Lewis, 1990), which can then guide the design of that interface. In cybersecurity, the general goal is to monitor and protect the cyber network (Gutzwiller et al., 2015). When protecting information (i.e., anything that occurs within a computer), operators in cybersecurity settings are concerned with confidentiality, integrity, and availability, where information cannot be disclosed, modified, or made private without authorization (Sawyer & Canham, 2019). Intentions and actions are formed, specified, and executed according to these goals, and afterwards, operators observe the difference in system state to see if an attack was successfully prevented. Examples of intentions and consequent actions are developing security patches, network monitoring for any potential security attacks, and identifying those attackers (Gutzwiller et al., 2015).

The stages-of-action theory utilizes constructs such as usability, user interface, and human error as users move between goals, intentions, and actions (Norman, 2013). One design principle from the theory pertains to users receiving constant feedback. Lack of adequate and/or pleasurable feedback for cybersecurity workers has been identified as a concern (McNeese et al., 2012), and therefore designers should more carefully take into account the latter set of user actions (i.e., perceiving the system state and onwards) that focus on the feedback and outcome of systems. To be more precise, cybersecurity systems do not often show operator impact after they prevent an attack. Instead, operators are simply rewarded with additional security work (e.g., more attacks to prevent and more patches to create; Gutzwiller et al., 2015). The feedback that is received also tends to be not clear and immediate enough, which can lead to misinterpretations due to cognitive biases and/or illusions (Einhorn & Hogwarth, 1978). For instance, if feedback overly focuses on potential attacks that could possibly be actual attacks, it fails to consider the number of attacks that can be missed, which creates an illusion of confidence (Gutzwiller et al., 2015). The lack of rewarding feedback can lead to frustration and boredom with cybersecurity vigilance tasks, which will eventually lead to burnout (Hancock & Warm, 1989). As there are also not many operators involved in cybersecurity work in the first place (Gutzwiller et al., 2015), extra care should be placed into designing hedonomic feedback on interfaces for the operators who are present.

When designing feedback for cybersecurity systems, designers can consider incorporating knowledge of results (KR) to inform operators about their response (Szalma et al., 2006). In cybersecurity settings, operators could at least be informed that they successfully thwarted a certain number of attacks yet also had a certain number of false alarms, as KR will ideally allow them to be more sensitive to detecting actual signals versus non-signals. However, while KR has been found to make users more conservative in responding “yes” (i.e., a signal was present) and therefore increased the reliability of that response, it worsened user ability to respond “no” (i.e., a signal was not present) or detect misses. Therefore, if KR is to be used in cybersecurity monitoring settings, the type of training provided should be considered as well, as effectiveness of KR will depend on whether emphasis is placed on reducing either the number of false alarms or misses.

As with all micro-HCI theories, the stages-of-action theory is more focused on the UI and measurable performance rather than usage context and changes in user behavior over time. Therefore, the stages-of-action theory would be more useful for a nondiscretionary cybersecurity system, as context and increasing level of expertise would likely not be issues—operators will use a specific system for their work and receive the necessary training (Grudin, 2005). However, if a cybersecurity system is being used in a more discretionary setting, where there are novice operators and/or users who require training or are from diverse backgrounds, then macro-HCI theories need to also be applied.

Essentially, while stages-of-action theory does help address the lack of feedback in cybersecurity, it does not address other issues relating to differing levels of expertise, work environments, and so on. Additional micro-HCI theories, such as consistency, should be used alongside stages-of-action theory to improve learnability of the interface and allow all types of users to carry out their tasks (Shneiderman et al., 2016), and macro-HCI theories should be used as necessary depending on specific usage contexts. Stages-of-action theory can be used as a starting point for cybersecurity design by creating an outline of user behavior and addressing the existing feedback issues, but additional HCI theories must be utilized to guide design as other factors can influence user experiences.

## Dynamic Theory

 Macro-HCI research takes an ethnographic approach of user interaction with the system over days or months. The dynamic macro-HCI theory addresses design for the gradual development of skills mastery, behavioral change, reputational growth, and capacities of leadership (Shneiderman et al., 2016). Users adopt new products with time, and conscientiously, there are five attributes influencing dynamic macro-HCI design guidelines (i.e., relative advantage, compatibility, less complexity, trial-ability, and observability; Rogers, 2003). These five attributions help determine the user’s adoption of a product.

Relative advantage, in cybersecurity, compares the innovation of the current software to the one preceding it (Walter, 2015). Compatibility is the introduction of the innovative system on current methods of cybersecurity. The ease of use of the innovation will impact the innovations’ adaptability, as a complex system will be more difficult to adopt. The testability and exploration possibilities the user can take from the innovated system will foster a larger acceptance rate. It is critical as it gives users the opportunity to experience and become aware of what the system’s innovations are capable of; giving the users an opportunity for a test run with the system and allowing for familiarity. Observability in cybersecurity refers to the level of exposure in the market of the innovative system. While dynamic theory is not applied very often in the development of cybersecurity software, it is important to understand who has used the system in an applied setting and how are the attributes defined.

# **Key Issues in Design**

## Complexity

There are many issues in the design of cybersecurity interfaces that need to be addressed. One of the major issues relates to the complexity of cybersecurity tasks. According to Nurse et al. (2011), cybersecurity tasks are extremely complex, which places extensive demands on the user that must be considered in the design process. For example, complex cybersecurity tasks may reveal limitations in human memory, such as failure to recall the standards and procedures associated with a particular system (Sasse et al., 2001). In some cybersecurity tasks, the operator must monitor the system in case of any breaches. These attacks may happen at any given time which requires the operator to always be active and alert. While performing this monotonous task, it is relatively uncommon that an attack will happen. Because attacks rarely occur, vigilance is a major factor in which being able to effectively discover attacks decreases as time goes on (Sawyer & Hancock, 2018). Furthermore, cybersecurity experts are tasked with considering the various types of attacks that can occur, as well as the location that a breach could take place. Effective cyberattacks are often difficult to notice and may be unknown for a long period of time (Gutzwiller et al., 2015). Being unaware of an attack is very problematic as extensive damage can be done before there are any measures taken to counter the attack. Therefore, the operator must anticipate attacks to reduce the consequences that may occur, further adding to the complexity of the task (Nurse et al., 2011). Additionally, most humans are not cybersecurity experts and lack the knowledge to properly understand the complex tactics used by hackers and the risks involved in a breach. This lack of knowledge tends to result in poor cybersecurity practices as individuals do not present the motivation to take extra precautions needed for these complex tasks (Boyce et al., 2011). In designing cybersecurity interfaces, these crucial aspects in the complexity of the task must be examined to promote successful interaction.

## Authentication

Authentication is the mechanism for different types of incoming requests and other types of information that provide identifying credentials that are compared to authorized ones (Daniel Ani et al., 2016). In system design, authenticity provides a roadblock for which processes can automatically be validated without interference from external sources. By not providing appropriate authentication, the confidentiality, integrity, and availability of system commands or the system as a whole can be compromised (Maple, 2017). For example, a system that uses Distributed Network Protocol-3 (DNP-3) and devices that were designed without security capabilities are more likely to be ignorant in differentiating between the system commands and external commands. This is because a system that uses DNP-3 cannot authenticate the “correct” system action and devices that were not designed with security capabilities have no way to check which users are authorized to perform actions (Daniel Ani et al., 2016). The introduction of mobile devices brought new issues to authentication through privacy and anonymity as authenticating mobile devices is difficult and almost all mobile devices are created differently, which requires authentication-specific software for each device. Though some measures have been implemented that help in authenticating user information, such as password pairs and two-step verification, not all of these processes can be established for every type of login on independent devices (Maple, 2017; Al-Muhtadi et al., 2017). Therefore, improvement in authenticating system commands and protocols is crucial to maintaining critical system functioning without external intervention, prolonging the event in which critical systems functions can occur by providing a safety net and source for validating information.

## Updates

Updates are another issue in cybersecurity because not providing the user with the latest improvements and changes puts their information at risk and may bring new vulnerabilities to the system. Additionally, updates to systems have to be automatic, easy to implement, free, and bring no negative side effects to the user’s experience (Maple, 2017). Updates need to be automatic so that users are always on the latest software to prevent data breaches to personal information, system commands, or another system component. Updates need to be cheap so that all users have equal access. This is further necessary for devices that are either connected to multiple other devices (i.e., “internet of things”) or are isolated systems, since connected devices can spread malicious information to one another and in between isolated systems. Therefore, initial internal testing has to be established to reduce the chances of any other part of the system being affected or the infrastructure being changed (Daniel et al., 2016). Lastly, any update that is implemented has to bring no severe changes to the user experience, since having an extreme change may cause users more confusion than solutions. Updates are essential since changing some parts of a running system allows for bug fixes and system recoveries can be performed fostering systems properties such as availability and malleability (Bannò et al., 2010). Due to these reasons, cybersecurity interfaces must be designed to account for better use and control over device authentication and system updates.

# **Practical Applications**

## Google Authenticator - Complexity

Google Authenticator addresses the issues related to task complexity in numerous ways by creating a simplistic way to receive a one-time password. First of all, the app allows for an easy initial setup as a quick response (QR) code can be used to link an account. This design aspect reduces the complexity of the task, resulting in decreased workload required to interact with the system (Sasse & Flechais, 2005). Once an account is linked, the app lists all the one-time passwords on the first page. This follows the flexibility and efficiency of use heuristic, as the user can quickly access any required password (Nielsen, 1994). Furthermore, the user can copy a one-time password by simply pressing on it. This method allows for an effortless transfer of the password from the authenticator to the desired account on the same device, which reduces the need for short-term memory of the code and saves cognitive resources. Having to recall passwords for multiple systems also increases the complexity of the task as it is very demanding on a user’s memory (Sasse et al., 2001). In Google Authenticator, there are no additional demands placed on the memory of a user as one-time passwords do not need to be remembered for long periods of time.

## Google Authenticator - Authentication

Google Authenticator addresses user authentication by providing users with one-time passwords or code that allows them access to their account if they are accessing it from a new or different device. Two-step verifications is one measure that Google implements and has shown to reduce the chances for hackers or other users to steal information or take over accounts and provides improved security for the authentication process (Seta et al., 2019). This adds a layer of security for users by providing specific codes or verification formats that only the user will know. Additionally, providing users with two-step verification has shown to be a cost-effective method to reduce the impersonation of users when accessing online information (Awasthi, 2015). By providing user authentication, Google can give users access to phone numbers, messages, register devices, and provide devices an extra layer of security.

## Google Authenticator - Updates

Google Authenticator addresses updates in a few ways. If users choose to use the online version of Google, then Google Authenticator will update automatically or through the update of Google Chrome. Users also have the option to download an app on their device that will allow them to become verified when using new devices, and this app is updated by Google through the Google Play app store. Through the given updates, users have been able to add additional devices to their account to minimize two-step verification on multiple devices, importation of new accounts, exportation of Google accounts and the addition of various safety features. Google and Google authenticator address updates by having them be automatic, easy to implement, and most importantly, they are free (Maple, 2017). Updates to Google Authenticator provides users with better ways to interact with their Google account and gives users updated protection for their vast connected array of information.

## Areas of Improvement and Recommendations - Google Authenticator

One aspect of Google Authenticator that remains in need of improvement relates to Nielsen’s first heuristic: visibility of system status (Nielsen, 1994). When the user opens up the app, they are presented with a six-digit code and a small, animated timer icon that runs on 30-second cycles. However, when examined more closely, users will find that the timer runs regardless of whether the app is open, and there is no visible feedback after a code has been used or has expired. As lack of adequate feedback has been previously identified as an issue in cybersecurity (Gutzwiller et al., 2015), Google Authenticator should better communicate system status to users. As a starting point, an explanation about time-based one-time passwords (TOTP) could be provided to users in the introduction and “How it works” section of the app, informing users that the 30-second timer runs according to the current time on the device (Sudar et al., 2017) and therefore will continue to count down even when the app is not open. Additionally, the users could be provided visible feedback each time the code is used or has expired (e.g., “Previous code has expired. Please enter the following code.”), as the app currently continuously presents codes on a 30-second cycle without any feedback.

When Google Authenticator fails to provide any feedback after the user utilizes the authentication code or the code expires, there is a chance the user may enter an incorrect code. Users may find that even if they enter a code late (e.g., halfway through the timer of the next 30-second interval), the code still works. This discovery can lead to confusion, as it is not clear which code the user should be using in Google Authenticator. While the codes are memorable, as they are short and visibly chunked in pairs of three-digit numbers (Miller, 1956), there is still the possibility of proactive interference (Engle, 2002) where the previous code could interfere with entering the new code. Therefore, the aforementioned adequate feedback could also assist in error prevention (Nielsen, 1994) as it will clarify which code the user should enter. In addition to including a message that says, for example, “This code has expired. Please enter the following code within the next 30 seconds” the old code should promptly be disabled so that only one code is active at a time.

Given that 35.72% and 64.28% of users worldwide either have their phone stolen or misplaced (Poggi & Ortega, 2020), respectively, Google Authenticator should include easy-to-find help documentation (Nielsen, 1994) to address these issues. However, the current interface does not provide this information—there is an option to transfer codes from the current device onto a new device, but otherwise no conspicuous “recover codes from a lost device” or equivalent. Additionally, none of the popular help pages (i.e., “Fix common issues with 2-Step Verification,” “Sign in if you lost your security key,” or “Set up a recovery phone number or email address”) give clear, step-by-step documentation to recover authentication codes from a lost device. Given that users could lose their device at any time, the issue should be clearly addressed and documented in the “Help & feedback” section, and possibly already available on the home page of the interface or in the introduction of the app (e.g., “Lost your previous device?).

## Recommendations - Artificial Pancreas

The artificial pancreas (AP) is a continuous glucose monitoring system that utilizes a controlled computer algorithm to automate the calculation of insulin doses and insulin delivery; providing tight blood glucose regulation in patients (Cobelli et al., 2011; Russell & Beck, 2016; Ramkissoon et.al, 2017). The Diabetes Technology Society introduced the Cybersecurity Standard for Connected Diabetes Devices in 2014 (Klonoff, 2015); creating a framework specifying security requirements of the devices and how to assure these requirements are met (Blauw et. al, 2016).

Do-it-Yourself (DIY) Hacking has become a prevalent issue with diabetes devices, where the devices are hacked by unauthorized agents and the patients themselves to extract data that is not readily available (Klonoff, 2015). There is a dilemma between having more access to data versus the need to protect the data from unauthorized access; DIY hacking movements intend to obtain data across multiple devices and better data visualization. This patient empowerment movement increases the risk of unauthorized access to confidential data.

Klonoff (2015) recommended the following five features for to strengthen the cybersecurity in the AP system, (1) continuous access to the system; (2) information confidentiality; (3) integrity without data alteration; (4) privileged access identification; and (5) identity verification before executing a task. It is recommended to specify the requirements regarding cryptography, secure and authorized communication, and integrity protection of software data (Blauw et.al., 2016) while allowing simplified but secure access to authorized patients and professionals. The AP system interface should have visible system status of the different components as well as the actions needed; adequate help and documentation about the system, data, and failures made accessible to the authorized user.

Signal detection theory assumes that detection is the sensitivity plus the response bias or the absolute threshold (Peterson et al., 1954); if the AP systems consist of too many alarms there needs to be a discriminating (difference) threshold. Having too many alarms that are not perfectly discriminable leads to error in alarm detection and response. Major alarm signals can be ignored, leading to health implications if not addressed promptly. Feedback and discrimination of signals should be distinguishable when there are several alarms present such as difference in sound, visuals, and vibration (Ramkissoon et. al., 2017). Visibility of system status is imperative for adequate signal detections. Lack of proper feedback has been linked to cybersecurity issues, users should be provided with a combination of discriminable signals and feedback to reduce error and risk of DIY hacking. Additionally, help and documentation should be available and accessible for proper maintenance of AP devices, and direction in case of system failures.

## Recommendations - Hawaii Emergency Management Agency Interface

The recommendations relating to visibility of system status through adequate feedback, error prevention when presented with multiple choices, and clear help and documentation for common potential issues can be applied to the interface utilized by the Hawaii Emergency Management Agency (HI-EMA) in 2018. In that year, one employee from the HI-EMA selected the actual ballistic missile alert to be sent out instead of a drill. On the interface, the options were presented in a drop-down menu which can be error prone as choices are near each other (Norman, 2018). Additionally, the interfaces had not been updated, and the file names for the drill and actual ballistic missile alert options were very similar (Medema et al., 2018) and had poor syntax (Norman, 2018). Error prevention would ensure that the options for drills and actual alerts in the system are clearly distinguishable from one another, which would assist in preventing employees from choosing the wrong option. Norman (2018) made further suggestions for error prevention, including having separate “test” and “live” modes for the system.

After the false alert was sent out, it took 38 minutes to send out a second alert as a correction. The delay in sending a correction was partially due to conflicting information between the HI-EMA and Federal Emergency Management Agency (FEMA), where the HI-EMA mistakenly believed they needed permission from the FEMA to issue a second alert (Medema et al., 2018). Other factors included HI-EMA not providing employees with clear actions to take after issuing a false alert, as well as having a vague checklist for missile alerts that was up to operation interpretation. Having clear help and documentation included in the interface when an incorrect alert is sent out in addition to proper communication between agencies could have helped decrease the delay in sending the correct second alert. A detailed and easy-to-understand checklist would have also helped by either preventing the false alert or quickly allowing operators to correct their mistake.

Although the HI-EMA interface did ask for confirmation before sending the alert (Medema et al., 2018; Norman, 2018), there could have been better communication of system status in the message box. According to Norman (2018), the confirmation dialogue did not state the actual message that was to be sent out—merely that some alert would be sent. He suggests that the consequences of sending out the message be clearly stated in that request for confirmation. Providing feedback afterwards, such as, “Alert successfully sent. [insert message] will be broadcasted” would help operators confirm whether they performed the action they intended. In combination with the aforementioned error prevention and help documentation, this type of feedback would reduce the chances of operator error and/or assist in quickly correcting an error on the HI-EMA interface.

# **Appendix A - Recorded Consultation Session Access**

Meeting Recording:

https://csulb.zoom.us/rec/share/V7Al8TbFO49C4FPAJ5SnJu6EZpU87ofxhlZfVNjaR-INYyRI4SOnR898mRd63dFs.MLxA0A\_ye9fyNkSE

Access Passcode: s9yX9FE+

# **References**

Al-Muhtadi, J., Shahzad, B., Saleem, K., Jameel, W., & Orgun, M. A. (2019). Cybersecurity and privacy issues for socially integrated mobile healthcare applications operating in a multi-cloud environment. *Health Informatics Journal, 25(2)*, 315–329.<https://doi.org/10.1177/1460458217706184>

Awasthi, A. (2015). Reducing identity theft using one-time passwords and SMS. The EDP Audit, Control, and Security Newsletter, 52(5), 9-19.<https://doi.org/10.1080/07366981.2015.1104935>

Bannò, F., Marletta, D., Pappalardo, G., & Tramontana, E. (2010). Tackling consistency issues

for runtime updating distributed systems. *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Ph.D. Forum (IPDPSW)*, pp. 1-8. https://doi.org/10.1109/IPDPSW.2010.5470863.

Boyce, M. W., Duma, K. M., Hettinger, L. J., Malone, T. B., Wilson, D. P., & Lockett-Reynolds,

J. (2011). Human performance in cybersecurity: A research agenda. *In proceedings of the Human Factors and Ergonomics Society 55th Annual Meeting*, *55*(1), 1115-1119.

Blauw, H., Keith-Hynes, P., Koops, R., & DeVries, J. H. (2016). A review of safety and

design requirements of the artificial pancreas. *Annals of biomedical engineering*, *44*(11), 3158-3172.

Cobelli, C., Renard, E., & Kovatchev, B. (2011). Artificial pancreas: Past, present,

future. *Diabetes*, *60*(11), 2672-2682.

Daniel Ani, U.P., He, H., & Tiwari, A. (2016). Review of cybersecurity issues in industrial critical infrastructure: Manufacturing in perspective. *Journal of Cyber Security Technology, 1(1*), 32 – 74.<https://doi.org/10.1080/23742917.2016.1252211>

Einhorn, H. J., & Hogarth, R. M. (1978). Confidence in judgment: Persistence of the

illusion of validity. *Psychological Review*, *85*(5), 395–416. <https://doi.org/10.1037/0033-295X.85.5.395>

Engle, R. W. (2002). Working memory capacity as executive attention. *Current Directions in*

*Psychological Science*, *11*(1), 19–23.<https://doi.org/10.1111/1467-8721.00160>

Grudin, J. (2005). Three faces of human-computer interaction. *IEEE Annals of the*

*History of Computing*, *27*(4), 46–62.<https://doi.org/10.1109/MAHC.2005.67>

Gutzwiller, R. S., Fugate, S., Sawyer, B. D., & Hancock, P. A. (2015). The human

factors of cyber network defense. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *59*(1), 322–326.<https://doi.org/10.1177/1541931215591067>

Hancock, P. A., & Warm, J. S. (1989). A dynamic model of stress and sustained

attention. *Human Factors*, *31*(5), 519–537. <https://doi.org/10.1177/001872088903100503>

Klonoff, D. C. (2015). Cybersecurity for connected diabetes devices. *Journal of Diabetes*

*Science and Technology*, *9*(5), 1143-1147.

Maple, C. (2017). Security and privacy in the internet of things. *Journal of Cyber Policy, 2(2*), 155 – 184.<https://doi.org/10.1080/23738871.2017.1366536>

McNeese, M., Cooke, N. J., D’Amico, A., Endsley, M. R., Gonzalez, C., Roth, E., &

Salas, E. (2012). Perspectives on the role of cognition in cyber security. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *56*(1), 268–271.<https://doi.org/10.1177/1071181312561063>

Medema, H., Blackman, H., Savchenko, K., & Boring, R. (2018). Human reliability

considerations from the Hawaii ballistic missile alert. *Probabilistic Safety Assessment and Management PSAM 14*, 1–9.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity

for processing information. *Psychological Review*, *63*(2), 81–97.<https://doi.org/10.1037/h0043158>

Miron, W. R. (2015). *Adoption of Cybersecurity Capability Maturity Models in*

*Municipal Governments* (Doctoral dissertation, Carleton University).

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Proceedings of the*

*SIGCHI Conference on Human Factors in Computing Systems*, 152–158.<https://doi.org/10.1145/191666.191729>

Norman, D. (2013). *The design of everyday things: Revised and expanded edition*.

New York, NY: Basic Books.

Norman, D. (2018). *What went wrong in Hawaii, human error? Nope, bad design*. Fast

Company. <https://www.fastcompany.com/90157153/don-norman-what-went-wrong-in-hawaii-human-error-nope-bad-design>

Nurse, J. R. C., Creese, S., Goldsmith, M., & Lamberts, K. (2011). Guidelines for usable

cybersecurity: Past and present. *In 2011 Third International Workshop on Cyberspace Safety and Security (CSS), 3*(1), 21-26.

Poggi, N., & Ortega, J. (2020). *Mobile theft & loss report 2020*. Prey Inc.

<https://preyproject.com/resources/ebooks/mobile-theft-report-2020>

Polson, P. G., & Lewis, C. H. (1990). Theory-based design for easily learned interfaces. *Human-*

*Computer Interaction*, *5*(2), 191–220.<https://doi.org/10.1207/s15327051hci0502%263_3>

Ramkissoon, C. M., Aufderheide, B., Bequette, B. W., & Vehí, J. (2017). A review of

safety and hazards associated with the artificial pancreas. *IEEE reviews in biomedical engineering*, *10*, 44-62.

Rogers, E. M. (2010). *Diffusion of innovations*. Simon and Schuster.

Russell, S. J., & Beck, R. W. (2016). Design considerations for artificial pancreas pivotal

studies. *Diabetes Care*, *39*(7), 1161-1167.

Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the weakest link: A

human/computer interaction approach to usable and effective security. *BT Technology Journal, 19*(3), 122-131.

Sasse, M. A., & Flechais, I. (2005). Usable security: Why do we need it? how do we get it? In L.

F. Cranor & S. Garfinkel (Eds.), *Security and usability: Designing secure systems that people can use* (pp. 13-30). Sebastopol, US: O’Reilly.

Sawyer, B. D., & Canham, M. (2019). Neurosecurity: Human brain electro-optical signals as

MASINT. *American Intelligence Journal*, *36*(2), 40–47.

Sawyer, B. D., & Hancock, P. A. (2018). Hacking the human: The prevalence paradox in

cybersecurity. *Human Factors, 60*(5), 597-609. <https://doi.org/10.1177/0018720818780472>.

Seta, H., Wati, T., & Kusuma, I.C. (2019). Implement Time-Based One-Time Password and Secure Hash Algorithm 1 for Security of Website Login Authentication. *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pp. 115-120, doi:10.1109/ICIMCIS48181.2019.8985196.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos,

N. (2016). *Designing the user interface: Strategies for effective human-computer interaction* (6th Edition). Hoboken, NJ: Pearson.

Shneiderman, B. (2011). Claiming success, charting the future: micro-HCI and macro-

HCI. *interactions*, *18*(5), 10-11.

Sudar, C., Arjun, S. K., & Deepthi, L. R. (2017). Time-based one-time password for Wi-Fi

authentication and security. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1212–1216.<https://doi.org/10.1109/ICACCI.2017.8126007>

Szalma, J. L., Hancock, P. A., Warm, J. S., Dember, W. N., & Parsons, K. S. (2006). Training

for vigilance: Using predictive power to evaluate feedback effectiveness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *48*(4), 682–692.<https://doi.org/10.1518/001872006779166343>