Bridging the Trust Gap: Integrating Models of Behavior and Perception

Raquel Hill
Department of Computer Science
School of Informatics and Computing
Indiana University
Bloomington, IN 47405
+1 812-856-5807
ralhill@indiana.edu

Devan Ray Donaldson
Department of Information and Library Science
School of Informatics and Computing
Indiana University
Bloomington, IN 47405 USA
+1 812-855-9723
drdonald@indiana.edu

ABSTRACT

In this paper, we propose a process-oriented trust framework that integrates an integrity-based trust model with the requirements and perceptions of those who manage and administer computing infrastructure. This integration enables a feedback loop between the system administrator and established models of trust that have been proposed to harden and secure systems. The proposed study will engage administrators in the design and use of mechanisms for establishing and evaluating the trust of cyberinfrastructure. The proposed study addresses a gap in current security research, which often views users as managers of a single computer, and not as an administrator of large computing environments. This work seeks to capture system administrators' perceptions of security and trust and incorporate real-world practices into the design of mechanisms for securing systems.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection.

General Terms

Trust, Measurement, Performance, Design, Security, Human Factors.

Keywords

Trusted Platform Module, Trusted Computing, User Trust Requirements.

1. INTRODUCTION

Trust is central to relationships among people, technologies, and organizations in society. When people provide sensitive information about themselves to organizations, they trust those organizations to protect that information from going into the wrong hands. From an access control infrastructure perspective, people expect that only the appropriate entities will have access to their sensitive information, and that organizations have mechanisms in place to ensure that unauthorized entities cannot gain access. Unfortunately, recent security breaches at Anthem® and Target® demonstrate that the level of trust people place in organizations that are responsible for handling their sensitive information can be undeserved [1; 19]. The importance of trust is perhaps best illuminated by considering the consequences of loss of trust. For example, after their security breach, Target® estimated that, they lost upwards of $148 million and expected earnings to drop to 78 cents a share from 85 cents a share, reflecting more cautious consumer spending [1].

Given the proliferation of malware and software bugs, that often leave our computing systems in a compromised state, we are often making a leap of faith that the computing application or system will perform as expected. Even with advanced security features of operating systems that require the verification of digital signatures before software is loaded, these systems are still vulnerable to compromise when private keys are stolen and digitally signed malware makes its way into a system. Recent reports state that there has been an increase in digitally signed malware since Stuxnet [5; 32]. Furthermore, while code signing may be effective in indicating the identity of the producer of a piece of code when certificates have not been compromised, it is not an effective means for indicating software trustworthiness since automated verifiers may ignore the files when validating the signature [5].

Various approaches have been proposed in the literature to establish and assess trust in a system. One overarching theme of these approaches is to harden the system in order to limit the introduction of malicious software. The underlying premise is that systems must have a secure configuration before they can be trusted to detect and respond to threats and vulnerabilities [3]. Requirements for hardening and securing systems have been previously proposed [3; 6; 18; 21; 22; 24]. These requirements include, specifying secure configurations, and verifying the origin and the integrity of software. Such requirements establish a model for trust and lay the foundation on which the trusted computing paradigm was built. If we can harden systems, we can reduce the likelihood that vulnerabilities will remain undetected for long periods of time.

There are many unanswered questions related to our ability to establish, assess, and maintain trust of our cyberinfrastructure components, including: Why do compromises go undetected and persist for extended periods of time? How do system administrators (i.e., those responsible for implementing plans for managing and securing computing infrastructure) assess and measure trustworthiness? Are measures for hardening systems

NSPW '15, September 08-11, 2015, Twente, Netherlands © 2015 ACM.
ISBN 978-1-4503-3754-0/15/09…$15.00
DOI: http://dx.doi.org/10.1145/2841113.2841125

ready for wide-scale deployment (i.e., multiple machines within an organization)? Would they enable users who manage and secure systems to do so more efficiently and effectively? Would the trust benefits of such devices be perceived as such by these system administrators? To address these questions, we propose to evaluate: an organization's functional and non-functional requirements, the mechanisms that are being used to assess the trust of the system, the administrators' perceptions of trust, and how these perceptions affect how they manage and secure systems. For example, organizations that support instruction and training must also support dynamic changes in infrastructure components. These dynamic changes may limit the applicability and effectiveness of integrity-based trust mechanisms that are better suited for more static environments. It is not sufficient to uniformly apply mechanisms to protect sensitive data and hardware resources.

McCune et al. [21] propose a simplified model that would serve as a root of trust for a user. While designed with the user in mind, no current research assesses user perceptions of such a device, whether it satisfies users' trust requirements and is deemed usable by those who manage an organization's computing infrastructure. In addition, there is no research that has looked at the effects of such assessment mechanisms on system administrators' perceptions of trust. For example, are hardware modules, like trusted platform modules being deployed to assess the integrity of cyber-infrastructure?

In this paper, we present a new trust framework that integrates behavioral and perception-based trust models. The framework also accounts for administrators' trust requirements. To validate the model, we propose a scale development study aimed at understanding trust requirements and subsequently measuring the trust perceptions of individuals who manage and secure computing infrastructure.

2. BACKGROUND

Trust is "the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context" [16]. Trust can be further decomposed into two categories: social trust and behavioral trust. Social trust is the belief in the safety or goodness of something because of reputation, association, recommendation, or perceived benefit. Behavioral trust, on the other hand, is the belief in the safety or goodness of something because it attests or proves that its actions meet the trusted criteria. Siegrist and Cvetkovich [27] argue that when an individual lacks knowledge about a hazard, social trust of authorities managing the hazards determines perceived risks and benefits. Their results suggest that the lay public relies on social trust when making judgments of risks and benefits when personal knowledge about a hazard is lacking.

Traditional assessment mechanisms like recommendation, reputation, and referral are used to determine whether to trust an entity. These mechanisms may be effective when used in social interactions between humans or maybe even technology assisted interactions with a centralized system. These mechanisms, often manually executed, are far less effective when used to assess the trust of a distributed entity. To be effective, your assessment mechanism must determine whether you are communicating with the appropriate entity, whether that entity's software will behave as expected, and whether the underlying communication infrastructure is functioning properly. The problem of assessing trust is further complicated by the proliferation of malware, faulty software, and various other security attacks (i.e., spoofing, phishing/imposter systems, etc.).

In this work, we propose to look at the user as a manager and an administrator of the infrastructure, not as the consumer who depends on the infrastructure to be in a trustworthy state. Security researchers have regarded the user in three ways. 1) The most common is the user of the functionality that the system or infrastructure provides. 2) Less common is the description of a user as the creator or developer of software. 3) Even less common is evocation of users as managers of infrastructure.

Multiple security researchers have proposed models for establishing trusted systems. These researchers motivate the need for trusted systems in order to support user trust judgments [21; 22; 24; 26]. For example, McCune et al. [21] state that it is the user who may need to use one device to verify the trustworthiness of another device, and yet, cannot verify trustworthiness because both devices are dependent on other devices. They offer the iTurtle solution as a single verification device, "an unambiguous point from which trust originates for the user". In another example, Parno [24] explains that in order for a user to trust a computer with a secret, "a user needs some assurance that the computer can be trusted". Hence, it is the user who needs information about the state of the system and its nodes in order to judge the system as trustworthy. Extending this discussion to cloud computing environments, Schiffman et al. [26] state that "a cloud customer must be able to verify that the cloud's integrity has not been compromised and that it is functioning within the parameters necessary to satisfy the customers' security needs." In this research, the user is conceptualized as the consumer of the functionality that a system provides.

Less common is research that focuses on the user as the creator of systems and software. For example, Balebako et al. [4] engage developers in software development. Indirectly, trust plays a role for their study participants because they are making trust decisions by relying upon the functionality that is provided by the modules that they use.

Security researchers have long acknowledged the need to consider the user in designing for trust [10; 21; 22; 24; 26]. For example, Camp [10] argues that:

"*experts focus on the considerable technological challenges of securing networks, building trust mechanisms, and devising security policies. Although these efforts are essential, that trust and security would be even better served if designs more systematically addressed the (sometimes irrational) people and institutions served by networked information systems.*"

Though research regarding the trust perceptions of IT security professionals is limited, various studies have explored some aspects of their experiences [2; 7; 15; 29]. Sundaramurthy et al. [29] conducted long-term participant observation in order to understand analysts' work[1] and build tools that addressed their needs. The primary function of the analyst in this study differs from that of a system administrator who manages, maintains, and secures infrastructure. In addition, these researchers did not explicitly focus on the ways in which analysts' conceptualize trust in computing infrastructure. Botta et al. [7] conduct a field study of IT professionals with the objective to build theory about how

[1] The primary responsibility of the analyst in this study was to find matches in alert logs that would move the investigation of an alert forward.

they practice security management given their human limitations and the realities of their workplace. Their results suggest that the job of IT security management is distributed across multiple employees often affiliated with different organizational units. Furnell et al. [15] investigate the challenges that IT Security practitioners face in their organizations, including the interaction among human, organizational, and technological factors. The work identifies 18 challenges that can affect IT Security management within an organization. Albrechtsen et al. [2] explore the digital divide between users and security managers. In their study, security managers do not implement/administer the security plan, but are mainly concerned with the non-technical aspects of information security, such as developing documented systems, arranging awareness campaigns, and supporting decision-makers at the line management level. Their findings show that managers and users have limited interactions, which leads to divergent views on what security means.

While the work of Albrechtsen et al. [2] examine the security perceptions of security managers and users, there is a gap in the literature regarding the security and trust perceptions of system administrators. There is a need to respond to this gap with a study aimed at understanding and subsequently measuring users' (i.e., system administrators') trust perceptions and incorporating this information into an integrity-based model of trust. Closing this gap could illuminate the utility of trusted systems when widely deployed to harden and evaluate the integrity of cyberinfrastructure.

3. PROPOSED MODEL

To orient our research questions, we present an integrated trust framework. This framework uses a process-oriented approach to derive and define a requirements-driven trust model. The ultimate goal of this work is to deploy this model in real world environments.

Our notion of trust is derived from requirements for hardening/securing systems. We take these requirements and situate them within the context of trusted computing, which provides a means for establishing a root and foundation of trust, enforcing system integrity, and attesting to the system state. Our proposed model is comprised of two components that create a feedback loop by which information is shared the components.

Using a trusted computing approach to implement trust can require a complex set of interactions. McCune et al. [21] simplify the presentation of these interactions. Figure 1 expands this view and illustrates an implementation of an integrity-based trust model that uses a hardware component along with trusted hardware to create a trusted computing base.

Components A-G are derived from work by Harris and Hill [18], and comprise a trusted computing mechanism for enforcing integrity-based trust. Components H and I extend the mechanism by integrating user requirements and perceptions into the framework for establishing, measuring and assessing trust. (H) represents users' trust requirements. (I) represents users' perceptions of trust. The user in our system is the administrator who is responsible for managing, maintaining and securing the system. The arrow between H and I represents the relationship between H and I; users' perceptions of trust based upon users' trust requirements.

Components A through G of Figure 1 illustrate our initial trust model. A separate hardware component and trusted grub serve as the root of trust. Dark solid lines represent the extension of trust from trusted boot loader (A) to a trusted kernel enabled to measure any software that is loaded (B) to the Trust services that support system attestation and integrity verification (D) and finally (if the trusted computing base is established) the Application (E). In this example, the kernel mounts the compressed ISO (C) as indicated by a solid line prior to starting the Trust services. This image may contain additional tools for establishing and measuring trust that are not provided by the kernel. Dashed lines show components of the system that are to be evaluated to determine whether they meet the trust requirements of the system. Any new file added to the system would also have a dashed line. All software must meet the trust criteria in order for a trusted state to be established and private data to be unsealed (F). With the information in (G), remote attestation can occur with the peers of the network.

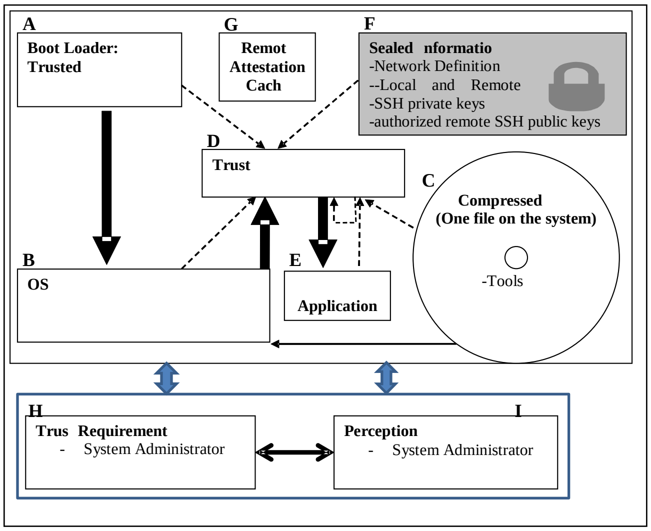

Figure 1. Process-Oriented Trust Framework

When the Trusted Computing Group's (TCG) Trusted Platform Module (TPM) [17] is used to implement Components A-G, these components enforce requirements for ensuring the trustworthiness of both the identity and the state of the system. These two requirements are essential for establishing secure communications.

The TPM, along with the Trusted Software Stack (TSS) provide mechanisms for creating unique cryptographic identities and for attesting a system's state using signed cryptographic hashes of files. Any system intending to utilize these features must ensure that trust metrics are extended from the hardware security mechanisms (secure BIOS/TPM) through the boot loader to the operating system (OS) and finally to the application requiring the trust relationship.

3.1 Bridging the Trust Gap

Components A through G in Figure 1 are effective in assessing and establishing trust when the software stacks of the communicating entities are assumed to be relatively static. This assumption does not hold for many cyberinfrastructure domains. When changes to the software stack are infrequent, administrators can specify the version of code to be executed for each software component, and deploy mechanisms that enforce those policies. Such mechanisms are less effective in dynamic environments, where software updates are applied without delay, and end-user systems execute varying configurations. Thus, there isn't a "one-size-fits-all" approach to assessing, establishing and maintaining trust within cyberinfrastructure.

The objectives of the proposed work are to: (1) capture the security and trust perceptions of system administrators so that we may understand how they characterize secure and trusted systems; (2) identify and bridge the gaps between the integrity-based trust model in Figure 1 and trust models that are being applied in real-world environments.

3.2 Trust Use Case

The proposed research is part of a larger project that explores what it means to trust cyberinfrastructure in different types of computing environments. Our initial use case is that of an academic department's computing environment that is tightly contained, but yet dynamic. We focus on issues of trust as they relate to managing and securing cyberinfrastructure within this environment.

The department's computing environment contains a diverse array of infrastructure components. Table 1 below provides details of the types of devices that are supported by the IT Staff.

	Servers	Virtual Machines	Work Stations	Mobile Devices(phones, laptops, tablets)
Linux	88	111	265	
Windows	6	16	395	107
Mac	4		176	165
iOS				89
Android				65
ChromeOS				2

Table 1: Departmental Infrastructure Summary

The department's infrastructure is distributed across five primary physical locations, including the University's Data Center. The Data Center is an F-4 tornado proof facility with backup power and chillers, on-site operations personnel, and 2-factor secure access. The academic department rents rack space from the university at the Data Center, including six physical servers and 17 virtual machines. These servers store the department's primary, critical, and sensitive data.

Figure 2 illustrates a high-level architecture of the department's infrastructure and its relationship with the University's infrastructure, including the Data Center, and other external entities, which we refer to as the "World". The outer circle represents the world. The first inner circle, representing the University, is illustrated with dashed lines, which indicates open ports that enable communication between the university and the outside world. The two smaller circles within the University denote the Department and the Data Center infrastructures. The overlap between the Department and the Data Center depicts the Department's critical data resources that are stored within the Data Center. The Data Center nodes can communicate without limitation with other nodes within the Data Center, but explicit firewall rules must be enabled to allow communication between the Department and the Data Center nodes.

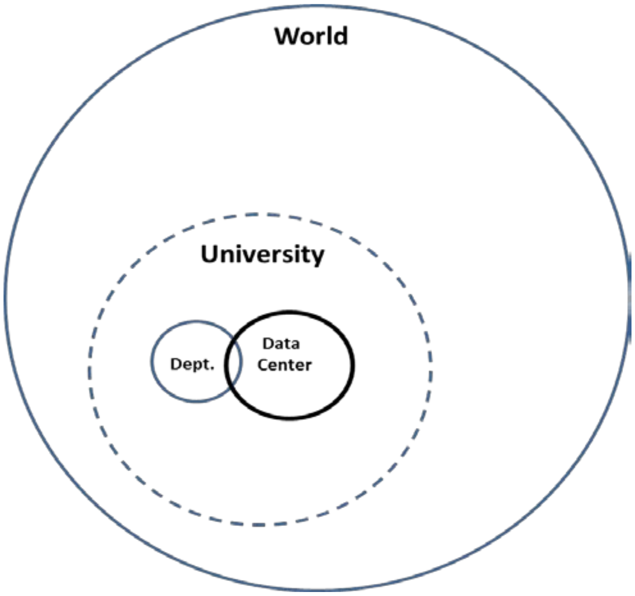

Figure 2: System Architecture

3.2.1 Trust Requirements for the Use Case

As previously stated, the trusted computing model, which relies on system hardening, may not be effective for dynamic environments with frequent changes to the software stack of infrastructure components. During our initial conversations with the Department's IT, we've learned the following:

- To facilitate research and creativity, the system must support dynamic changes to nodes, which further complicates the process of specifying software hardening policies;
- Functional services, such as instruction and experimentation, contribute to the dynamic nature of the environment and prevent IT staff from hardening some infrastructure components;
- Requirements to support external collaborators introduce infrastructure components that are outside of the administrative control of the IT staff;
- Given the dynamic nature of the environment, integrity-based trust mechanisms that do not allow a range of acceptable software versions may produce many false positives, rendering them ineffective for assessing the trustworthiness of dynamic components;
- Behavioral-based assessment mechanisms (e.g., anomaly detection) are often used to assess the trust of nodes in dynamic environments.

3.2.2 Characterizing Trust Requirement

In order to integrate the trust requirements and perceptions of trust of the system administrator into the trust model, we first need a framework for capturing these requirements and perceptions. In preliminary discussions with a lead system administrator, we observed that the administrator discussed protection mechanisms from the perspective of protecting against threats to the system and not necessarily from the perspective of satisfying a security requirement or goal. Unfortunately, the reality of managing systems is often reactive. Therefore, a more natural way of eliciting trust requirements from system administrators is to engage them in an exercise of threat modeling. During the threat modeling process, assets are identified; threats against the assets are enumerated; the likelihood and damage of threats are quantified; and mechanisms for mitigating threats are proposed. Myagmar et al. [23] champion threat modeling as a means of eliciting security requirements and offer that an initial set of security requirements can be obtained by converting threat statements into “shall not” requirement statements.

Quantifying the risk of threats enables administrators to rank the order in which threats should be addressed. Various approaches have been proposed for characterizing and quantifying the risk of threats, including calculating risk as the product of the damage potential and the likelihood of occurrence, *Risk = Criticality * Likelihood of Occurrence* [20]. Dread, an approach proposed by Microsoft, calculates risk across several categories, including: Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability [20]. Using Dread, a threat is rated on a scale from 1 to 10, for each category, with the resulting risk being the average of all ratings. Butler and Fischbeck [9] propose a multiple attribute threat index (TI) for assessing the risk of a threat. TI captures the relative importance of each type of threat [9], where $TI_a = Freq_a * (\sum_{j=attributes} W_j * X_{aj})$, and W*j* is the attribute weight and X*aj* is the most likely outcome value for the threat.

While quantifying risks will enable us to capture an organization’s security requirements, Pfleeger [25] advises that we should avoid false precision by doing the following:

- Base the probability distribution of a threat/attack occurring on historical data, not just on expert judgment;
- Since "both scientists and lay people may underestimate the error and unreliability in small samples of data, particularly when the results are consistent with preconceived, emotion-based beliefs”, we are to be mindful of the size of our experiments and the scalability of our results.

3.2.3 Assessing Mechanisms

While threat modeling enables us to capture requirements, a system administrator’s perceptions of the trustworthiness of a system are directly tied to her assessment of the effectiveness of mechanisms that have been proposed to mitigate threats. In addition to effectiveness, there is also the cost/benefit tradeoff of one mechanism versus another. If the cost of deploying and maintaining a mechanism exceeds the perceived benefit, its overall effectiveness will be rated/perceived as low. Therefore, the challenge of capturing perceptions of trustworthiness is providing a framework that enables the system administrator to compare mechanisms that have been proposed for mitigating the same threat.

We propose to leverage and extend Bulter’s [8] Security Attribute Evaluation Method (SAEM) framework. SAEM is a cost-benefit analysis process for analyzing security design decisions that involves four steps: 1) assessing the benefit of a security technology, 2) evaluating the effect of security technologies in mitigating risks, 3) assessing coverage security technology, and 4) analyzing cost. The SAEM process uses the previously defined threat index to characterize threats and potential attacks. The perceptions of security managers are used to quantify the benefit of security mechanisms. In an effort to avoid the pitfalls of quantifying perceptions that may vary broadly [25], we propose to augment perception ratings with historical data that quantifies the increase/reduction of the occurrence of attacks after a specific security mechanism has been deployed.

4. PROPOSED STUDY

In the previous section, we proposed a model that could simultaneously account for an integrity-based conceptualization of trust within cyberinfrastructure environments and a perception-based conceptualization of trust based on the trust requirements of those who manage the infrastructure. The integrity-based trust portion of the proposed model (i.e., the aggregation of letters A through G in Figure 1) has been previously validated [10]. On the other hand, the perception-based trust portion of the framework (i.e., the aggregation of letters H through I in Figure 1) has not yet been validated. Toward that end, we propose a scale development study to specify the components of H and I in Figure 1 and subsequently test the entire model (i.e., letters A through I in Figure 1). Specifically, we propose a study aimed at specifying perception-based trust in a way that is quite novel because it is grounded in the trust requirements and perceptions of system administrators.

To ascertain system administrators’ trust requirements and subsequently measure their perceptions based upon their requirements, we propose using the established methodology of scale development [12; 28]. The premise for proposing scale development for this study is that, if trust perception does in fact exist, then it ought to be definable and measurable. Although scale development has never been used to examine users’ trust in cyberinfrastructure, it is promising for this study because prior research demonstrates that users in other research domains are capable of articulating their trust requirements, and researchers are, in turn, capable of measuring users’ trust perceptions based upon their requirements [13; 14].

Scale development is a rigorous methodology consisting of four steps: 1) Construct Definition, 2) Generating an Item Pool, 3) Designing the Scale, and 4) Full Administration and Item Analysis (see Figure 2).

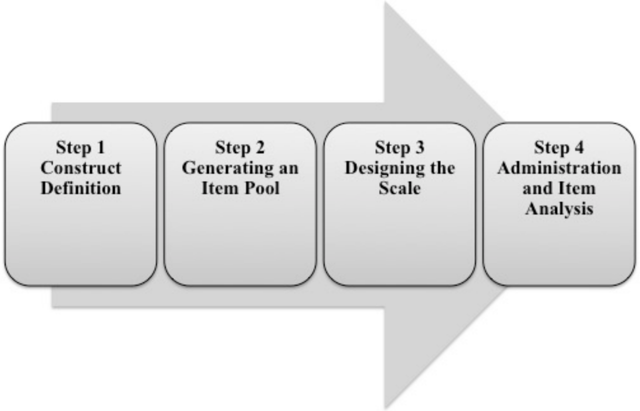

Figure 2. Scale development procedure adapted from Spector [28] and DeVellis [12].

Each of the four steps of scale development has its own set of methods. The methods of each step build on one another and, in turn, each step builds on the one preceding it. Hence the arrow pointing from left to right containing the steps of scale development in Figure 2. The remainder of this section describes the four steps of scale development in detail, including description of the methods that are necessary for execution of each step.

4.1 Step 1—Construct Definition

We will apply Step 1 of scale development—construct definition—in two ways. First, we will focus on clearly defining our goals for measuring trust. Second, we will conduct focus groups with a sample of real-world system administrators to understand their trust requirements. To engage the administrators, focus group questions will center on what security means to them, what it means to them for infrastructure to be secure, and what it means to them for infrastructure to be trustworthy (or in a trustworthy state). To further guide the discussion, the administrators will engage in threat modeling and mechanism assessment to elicit their trust requirements. The focus groups will conclude with discussion of and reactions to the integrity-based trust framework represented by Components A through G in Figure 1. Presenting this integrity-based trust model will be reserved until the end of the focus groups in an effort to avoid biasing the administrators' notions of trust before they get a chance to articulate what trust in computing infrastructure actually means to them.

4.2 Step 2—Generating an Item Pool

We will derive items for a trust perception measurement scale from participants' responses during focus groups that we will complete during Step 1 on the topic of their trust requirements. Specifically, we will derive items from the focus groups by turning statements that participants provide regarding their trust requirements into items for measurement of trust perception.

4.3 Step 3—Designing the Scale

During Step 3, we will transform the item pool resulting from Step 2 into a survey instrument for pretesting and refinement. Step 3 will involve seven activities. First, we will select stems for the items that are appropriate for the study (e.g., "The node is…", "The system is…", "The OS is…"). Second, we will select response categories and choices for the items. For example, we could use a 7-point scale ranging from -3 (very untrustworthy) to +3 (very trustworthy). We could also include the option -99 (not applicable) for participants to utilize if they feel they have no basis upon which to evaluate an item. Third, we will write instructions for participants regarding the survey. For example, the instructions will ask participants to rate how trustworthy they would perceive a node, given the trust requirement described in each item. Fourth, we will select web administration for the survey to make it easier for the study participants to take the survey and to ensure electronic data collection. Fifth, we will randomize the order of the items to avoid order effects. Sixth, we will pretest the survey by conducting cognitive interviews with a sample of participants. Seventh, we will revise the survey items and instructions based upon the cognitive interviewees' feedback.

4.4 Step 4—Full Administration and Item Analysis

During Step 4, we will administer the survey instrument resulting from Step 3 to a large sample of systems administrators via online survey software. We will recruit participants who have real-world experience managing infrastructures from national labs and academic institutions to take our survey.

To analyze the resulting data, we will perform two types of analysis: item analysis and exploratory factor analysis. Item analysis will involve analysis of item variances, item-total correlations, item means, and item standard deviations [12]. After performing item analysis, we will conduct exploratory factor analysis (EFA)[2] to establish the factor structure of the trust items [12]. We will operationalize important trust items as items with high factor loadings on factors with large eigenvalues.[3] What will not be immediately clear is exactly which items are the strongest indicators of trust. Consequently, we will use exploratory factor analysis as a tool to help identify the most critical items for measurement of trust.

We will perform EFA on the items that the participants evaluate. We will run EFA on the items using principal axis factoring with oblique rotation, as there is no reason why we would not expect whatever factors arise not to be correlated. We will rely upon results of Cattell's [11] scree test to determine the number of factors to retain. To assign items to factors, we will consider factor loadings equal to or higher than .32 [30].

In the remainder of this paper, we report the possible outcomes that our proposed framework is robust enough to capture. We also discuss the implications of what our study demonstrates regarding a user-oriented trust measurement strategy for inspiring new thinking in trust and security research.

5. POSSIBLE OUTCOMES

Table 1 illustrates the possible trust outcomes that our proposed model is designed to take into account. As shown in Table 1, there are six possible outcomes when considering integrity- and perception-based trust. The most favorable possible outcome (Y, Y) is listed first. In this circumstance, the cyberinfrastructure proves itself trustworthy (i.e., integrity-based trust) and users perceive that environment as trustworthy (i.e., users perceive the environment as aligned with their trust requirements or perception-based trust). The second, less favorable but still possible outcome is that the environment is verified as trustworthy from an integrity-based trust perspective (i.e., Y), but yet actual administrators do not perceive the environment as trustworthy (i.e., N). A third possible outcome is that the environment cannot be verified as trustworthy from an integrity-based perspective (i.e., N), and yet actual administrators perceive the environment as trustworthy (i.e., Y). Finally, the least favorable possible outcome is that the environment cannot be verified as trustworthy from an integrity-based perspective (i.e., N), and actual administrators also fail to perceive the environment as trustworthy (i.e., N).

[2] Exploratory factor analysis is a tool used to describe the correlation between observable variables (i.e., measurement items such as survey questions) and unobservable variables (i.e., constructs that are not directly available to the senses, in this case, trust perception).

[3] In factor analysis, it is assumed that the reason items correlate is because they are related to the same concept (i.e., they share a common factor). Factor loadings demonstrate the degree to which items correlate with factors. Eigenvalues are statistical measures of how much variance factors explain. The greater the eigenvalue, the more the factor reflects the concept under investigation, in this case, trust. Items with high factor loadings are the result of factors that contribute greatly to those items. For these reasons, we will analyze our data focusing on items with high factor loadings on factors with large eigenvalues. For more on exploratory factor analysis, see [12].

Table 1. Possible Trust Outcomes.

TRUST	
Integrity-based	Perception-based (i.e., administrators)
Y	Y
Y	N
N	Y
N	N

Although (Y, Y) is the most favorable possible outcome and (N, N) is the least favorable possible outcome, both are important because they represent correspondence between integrity-based and perception-based trust. Our trust model allows for comparison along these lines such that we can empirically examine the correspondence between these different forms of trust that we are bringing together in the context of our model.

(Y, N) and (N, Y) represent instances where integrity-based and perception-based trusts are out of synch. These instances are important as well and would require more research to examine the root cause(s) of disconnect between both forms of trust.

6. CONCLUSIONS

For nearly twenty years, researchers have called for a closer correspondence between technological security and trust research and research on the users of the advancements presented in such research [10; 31]. Few researchers have addressed this call. What is more common is that the concept of the user is often evoked as the beneficiary of the advancements described in security and trust research, but yet there is little research on users to verify the utility of the research findings for real-world users [21; 22; 24].

We take the next step by creating a feedback loop between system administrators and the mechanisms for establishing and evaluating trust. We propose to extend the current work in trust and security research by examining technological definitions of trust and security alongside users’ perceptions of those definitions as they apply to technology with which they actually interact.

To validate our results, we will deploy our survey to another sample of system administrators from national labs and academic institutions, conduct EFA on the resulting data, and compare results of the second run of EFA with results from the first run of EFA. If the results are similar with regard to the best performing trust items, we will include those items in a scale for measurement of trust perception. By design our proposed study will offer a much higher level of validation than what is typically reported in trust and security research.

7. ACKNOWLEDGMENTS

We would like to thank Rob Henderson for describing the departmental infrastructure, the mechanisms used to secure the infrastructure, and illuminating the challenges of establishing, assessing, and maintaining trust of cyberinfrastructure components.

8. REFERENCES

[1] Abrams, R., 2014. Target Puts Data Breach Costs at $148 Million, and Forecasts Profit Drop. In *The New York Times*.

[2] Albrechtsen, E. and Hovden, J., 2009. The information security digital divide between information security managers and users. *Comput. Secur. 28*, 6, 476-490. DOI= http://dx.doi.org/10.1016/j.cose.2009.01.003.

[3] Allen, J.H., 2001. Cert system and network security practices. In *Proceedings of the Fifth National Colloquium for Information Systems Security Education (NCISSE'01), George Mason University, Fairfax, VA USA*, 22-24.

[4] Balebako, R., Marsh, A., Lin, J., Hong, J., and Cranor, L., 2014. The Privacy and Security Behaviors of Smartphone App Developers. In *Proceedings of the USEC ’14* (San Diego, CA, 2014), Internet Society.

[5] Bencsáth, B., Pék, G., Buttyán, L., and Felegyhazi, M., 2012. The cousins of stuxnet: Duqu, flame, and gauss. *Future Internet 4*, 4 (November 2012), 971-1003.

[6] Biba, K.J., 1977. *Integrity Considerations for Secure Computer Systems.*

[7] Botta, D., Werlinger, R., Gagné, A., Beznosov, K., Iverson, L., Fels, S., and Fisher, B., 2007. Towards understanding IT security professionals and their tools. In *Proceedings of the Proceedings of the 3rd symposium on Usable privacy and security* (Pittsburgh, Pennsylvania, USA, 2007), ACM, 1280693, 100-111. DOI= http://dx.doi.org/10.1145/1280680.1280693.

[8] Butler, S.A., 2002. Security attribute evaluation method: a cost-benefit approach. In *Proceedings of the 24th international conference on Software engineering* ACM, 232-240.

[9] Butler, S.A. and Fischbeck, P., 2002. Multi-attribute risk assessment. In *Symposium on Requirements Engineering for Information Security*.

[10] Camp, L.J., 2003. Designing for trust. In *Trust, reputation, and security: Theories and practice* Springer, Berlin Heidelberg, 15-29.

[11] Cattell, R.B., 1966. The scree test for the number of factors. *Multivariate behavioral research 1* (2), 245-276.

[12] DeVellis, R.F. *Scale development : theory and applications*. SAGE, Thousand Oaks, Calif., 2012.

[13] Donaldson, D.R., 2015. Development of a scale for measuring perceptions of trustworthiness for digitized archival documents. Doctoral Thesis. University of Michigan.

[14] Donaldson, D.R. and Conway, P., 2015. User conceptions of trustworthiness for digital archival documents. *Journal of the Association for Information Science and Technology*. DOI= http://dx.doi.org/10.1002/asi.23330.

[15] Furnell, S.M., Clarke, N., Werlinger, R., Hawkey, K., and Beznosov, K., 2009. An integrated view of human, organizational, and technological challenges of IT security management. *Information Management & Computer Security 17* (1), 4-19.

[16] Grandison, T. and Sloman, M., 2000. A survey of trust in internet applications. *Communications Surveys & Tutorials, IEEE 3* (4), 2-16. DOI= http://dx.doi.org/10.1109/COMST.2000.5340804.

[17] Trusted Computing Group, 2015. Retrieved November 3, 2015, from www.trustedcomputinggroup.org.

[18] Harris, J. and Hill, R., 2010. Building a trusted image for embedded systems. In *Proceedings of the Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research* (Oak Ridge, Tennessee, USA, 2010), ACM, 1852739, 1-4. DOI= http://dx.doi.org/10.1145/1852666.1852739.

[19] Hiltzik, M., 2015, March 6. Anthem is warning consumers about its huge data breach. Here’s a translation. In *Los Angeles Times*.

[20] Howard, M. and Leblanc, D.E., 2002. *Writing Secure Code*. Microsoft Press.

[21] Mccune, J.M., Perrig, A., Seshadri, A., and Doorn, L.V., 2007. Turtles all the way down: research challenges in user-based attestation. In *Proceedings of the Proceedings of the 2nd USENIX workshop on Hot topics in security* (Boston, MA, 2007), USENIX Association, 1361425, 1-5.

[22] Moyer, T., Butler, K., Schiffman, J., Mcdaniel, P., and Jaeger, T., 2012. Scalable Web Content Attestation. *Computers, IEEE Transactions on 61* (5), 686-699. DOI= http://dx.doi.org/10.1109/TC.2011.60.

[23] Myagmar, S., Lee, A.J., and Yurcik, W., 2005. Threat Modeling as a Basis for Security Requirements. In *Proceedings of the IEEE Symposium on Requirements Engineering for Information Security (SREIS'05)* (Paris, France, 2005).

[24] Parno, B., 2008. Bootstrapping trust in a "trusted" platform. In *Proceedings of the Proceedings of the 3rd conference on Hot topics in security* (San Jose, CA, 2008), USENIX Association, 1496680, 1-6.

[25] Pfleeger, S.L., 2000. Risky business: what we have yet to learn about risk management. *Journal of Systems and Software 53* (3), 265-273.

[26] Schiffman, J., Moyer, T., Vijayakumar, H., Jaeger, T., and Mcdaniel, P., 2010. Seeding clouds with trust anchors. In *Proceedings of the Proceedings of the 2010 ACM workshop on Cloud computing security workshop* (Chicago, Illinois, USA, 2010), ACM, 1866843, 43-46. DOI= http://dx.doi.org/10.1145/1866835.1866843.

[27] Siegrist, M. and Cvetkovich, G., 2000. Perception of Hazards: The Role of Social Trust and Knowledge. *Risk Analysis 20* (5), 713-720. DOI= http://dx.doi.org/10.1111/0272-4332.205064.

[28] Spector, P.E., 1992. Summated rating scale construction: an introduction. SAGE Publications, Newbury Park, Calif.

[29] Sundaramurthy, S.C., McHugh, J., Ou, X.S., Rajagopalan, S.R., and Wesch, M., 2014. An anthropological approach to studying CSIRTs. *IEEE Security & Privacy*, 5, 52-60.

[30] Tabachnick, B.G. and Fidell, L.S. Using multivariate statistics. Pearson Education, Boston, 2013.

[31] Whitten, A. and Tygar, J.D., 1998. *Usability of Security: A Case Study*. Carnegie Mellon University.

[32] Yin, S., 2012. More Malware Discovered with Stolen Certificates. In *Security Watch*.