Developing a Requirements Framework for Cybercraft Trust Evaluation[†]

J. Todd McDonald[*], Shannon Hunt[*]
Department of Electrical and Computer Engineering,
Air Force Institute of Technology, Wright Patterson, USA
jmcdonal@afit.edu, shannon.hunt@us.af.mil

Abstract: It should be no surprise that Department of Defense (DoD) and U.S. Air Force (USAF) networks are the target of constant attack. As a result, network defense remains a high priority for cyber warriors. On the technical side, trust issues for a comprehensive end-to-end network defense solution are abundant and involve multiple layers of complexity. The Air Force Research Labs (AFRL) is currently investigating feasibility for a holistic approach to network defense, called Cybercraft. We envision Cybercraft to be trusted computer entities that cooperate with other Cybercraft to provide autonomous and responsive network defense services. A top research goal related to Cybercraft centers around how we may examine and ultimately prove features related to this root of trust. In this work, we investigate use-case scenarios for Cybercraft operation with a view towards analyzing and expressing trust requirements inherent in the environment. Based on a limited subset of functional requirements for Cybercraft in terms of their role, we consider how current trust models may be used to answer various questions of trust between components. In this work we characterize generic model components that will help answer questions regarding Cybercraft trust and pose relevant comparison criteria as evaluation points for various (existing) trust models. Our contribution in this research provides a framework for comparing trust models that are applicable to similar network-based architectures.

Keywords: network defense, trust, Cybercraft, trust model, requirements, threat model, attack tree

1. Introduction

The world is in the middle of the information age with almost anything easily accessible through the Internet. With this newfound freedom of information, there are many new threats fighting against the U.S. our military must think about. Working towards combating these new threats, the USAF redefined its mission in 2005 to "deliver sovereign options for the defense of the United States of America and its global interests - to fly and fight in Air, Space, and Cyberspace." (Gettle 2005) The strategic vision for Cyberspace as a warfighting domain was furthered by AF Secretary of Defense Michael W. Wynne when he announced the creation of a Cyberspace Command (AFCYBER) (Lopez 2006). Focusing on science and technology issues related to this domain, AFRL has launched a research initiative geared to prepare for defense in this critical realm, termed "Cybercraft". Just as we have aircraft platforms that operate in air and carry a wide variety of payloads (bombs, missiles, electronic warfare pods, precision guided munitions), the term Cybercraft reflects the idea of generic platforms that operate in cyberspace and execute a wide variety of payloads (patch verification, router configuration information, INFOCON policy enforcement, etc.).

The likely Cybercraft architecture consists of a machine-installed platform that executes mission-specific payloads: the platform represents a trusted component that provides command, control, and communication of cyber capabilities on host nodes while the payloads will inherit trust from the platform and carry out various defensive missions and goals. Network defenders will use this architecture to accomplish specific tasks via single or multiple cooperating Cybercraft payloads. The architecture will incorporate hybrid trusted hardware/software/firmware components in various levels of interaction to support desired functionalities such as intrusion detection, anti-virus monitoring, network defense, and so forth. Cybercraft may eventually be deployed on up to one million nodes in order to provide commanders with a root of trust related to their high level strategic and operational network defense needs.

The ability to model, measure, and verify the degree of trust a commander may place in Cybercraft remains a crucial research question that must be adequately characterized before this future architecture

---

[*] The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government

† This material is based upon work supported in part by the U.S. Air Force Office of Scientific Research under grant number F1ATA07355J001

becomes a reality. Although trust itself has a multitude of meanings related to security, in this work we consider how to synthesize and express the nature of trust in the future Cybercraft environment based on expected operational requirements. We further characterize generic model components that will help answer questions regarding Cybercraft trust and pose relevant comparison criteria as evaluation points for various (existing) trust models. We also introduce a novel approach to synthesize trust relationships iteratively based on use case analysis and attack tree threat modeling.

We structure the remainder of the paper as follows: we first present Cybercraft and trust issues in greater detail followed by presentation of our approach for trust distillation based on threat modeling. In the next section we evaluate relevant trust models and pose criteria for answering trust-based questions. Finally, we summarize our contributions and give recommendations for future work.

2. Related Work

Trust is an elusive concept that does not have a clear definition. One of the reasons for this is that trust is a social issue as opposed to technical. Although trust is used everywhere, Gollmann argues that just using the word ("trust") in a system or project is dangerous because of its manifold and sometimes contradictory meanings (Gollmann 2005). It is an overloaded term that hinders the clarity and precision that is sought after in technical fields. Nonetheless, it expresses a quality that military commanders make quite frequently: an objective dependability (whether by mathematical proof or demonstrated testing) that a system will perform according to its specifications, even though negative consequences can occur. We discuss next the specific ideas of trust that apply to Cybercraft in context to the envisioned architecture.

2.1 Trust

Many authors have attempted to define trust. Gambetta (Gambetta 2000) laid the foundation for the definition of trust as a social concept that is subjective and context-dependent. Cahill (Cahill 2003) elaborated further to add attributes such as self-preserving and self-amplifying, among others. In more general terms, trust is defined as the measure of trustworthiness that relies on whatever evidence is provided or implied (Bishop 2003). With this, trust is inherently linked with security and risk. When an entity is evaluating whether to trust another entity in a certain context, there is a specific risk the entity must be willing to take to form the relationship. Taking this risk will determine the security of the underlying system. If there is no trust, there is a lack of security. Trust plays a key role in system development and we consider it an essential concept and possible component of a security-based infrastructure.

To understand trust, many look at and try to mimic human trust (Cahill 2003, Capra 2004) and consider three main delineations: initial trust, trust evolution, and trust delegation. Initial trust is the first formation of trust between two entities and usually happens through recommendations from other trusted entities. Trust evolution is the continuation and self-adaptation of trust over time and allows for experience to affect the trust relationship. Trust delegation occurs when an entity delegates a trust decision to another trusted entity. In other trust domains (Ray and Chakraborty 2004), different terms express the same basic concepts: experience (for evolution), knowledge (initial), recommendations (delegation). Once we assign a precise meaning and definition for trust, we in essence form a model which may be exercised and evaluated given the assumptions and boundaries of our system. It is essential that regardless of the model chosen, the reason we want to use the model and our expectation of what it will provide in terms of security must be clearly defined.

Two entities can also have varying degrees of trust in each other, within a specific context. An entity can trust an individual in multiple contexts as well, each having a different value of trust. We can express transitive trust as the resulting measure between an entity A and entity C based on the assumption that entity A trusts entity B and entity B trusts entity C. Our interest in transitivity permeates a basic desire for Cybercraft: the ability to take a locality of guaranteed trust (established through hardware) and extend that trust to the execution of code (via payloads) so that the resulting effects, collected data, sensing information, and network operations are trusted as well. The other aspect of trust we must consider deals with the multi-agent communication and cooperation needs that become evident when considering a typical Cybercraft application involving multiple payloads operating across multiple platforms. Both of

these application contexts have ramifications for Cybercraft and our desire to express and measure commander-level trust.

2.2 Cybercraft

Phister et al. (Phister 2006) pose the first conceptual use of Cybercraft as an autonomous, intelligent agent that accomplishes military purposes across a wide variety of electronic-based media. Envisioned as a cyber-vehicle that traverses through cyberspace, Cybercraft were seen as the future platform by which military operations would be conducted in the cyber realm. AFRL enhanced these ideas (AFRL 2007) and developed platform/payload architecture with certain target qualities. The Cybercraft platform, for example, requires a long service life with large investment to support a variety of missions and will be the subject of intense scrutiny to characterize attribution, authentication, and reliability. The Cybercraft payload, likewise, supports rapid development cycles, provides extensibility, and implements specific effects related to defensive missions. The long service life of the platform allows for trust to be formed, maintained, and reevaluated on a constant basis. As research has progressed, trust and a self-protection guarantee for Cybercraft have emerged as a coherent study area.

We can use a domain model to describe various relationships between entities in a system. Figure 1 illustrates a rudimentary domain model for Cybercraft that illustrates several concepts pertinent to trust expression and security. First of all, the platform has a one-to-many relationship with prospective nodes, meaning that one platform will be deployed per node and nodes represent a wide variety of IP-based appliances such as workstations, routers, hand-held devices, or servers. Currently, Cybercraft platforms will execute payloads to achieve or accomplish specific effects and goals in support of operational/tactical missions. Platforms may communicate with other platforms or allow inter-payload communication to accomplish their tasks. Platforms (and thus payloads) may also use other tools or processes (virus checkers, IDS, etc.), depending on the level of trust such tools may have. Though the entire cyber environment is not represented, we can still visualize that nodes are connected to other nodes via networks and are ultimately controlled by some underlying operating system (OS). The specific relationship between the Cybercraft platform and a deployment node is still under consideration, but the current direction assumes a mixture of hardware and firmware that is independent of normal architectural layout.

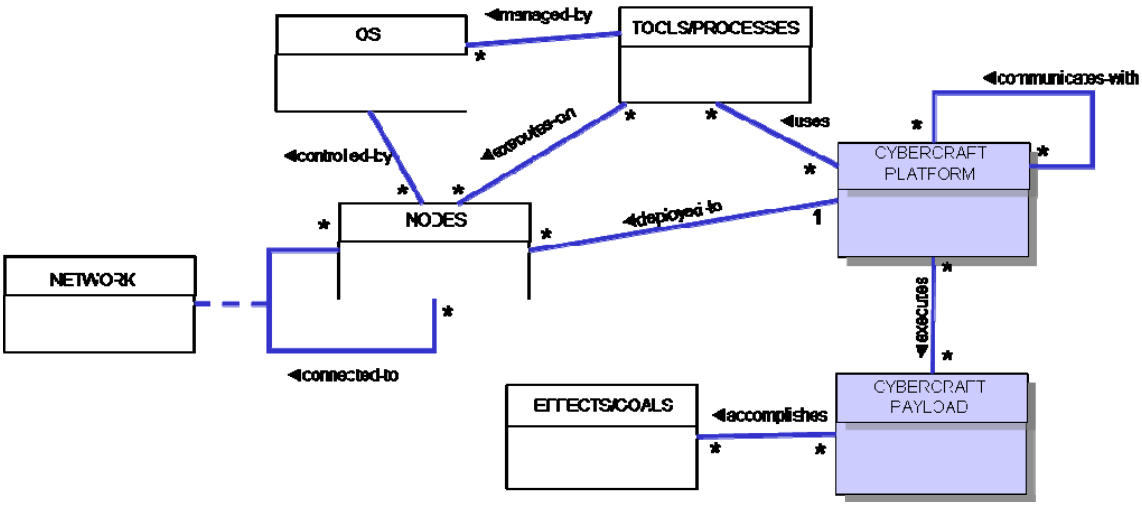

Fig.1: Domain model for Cybercraft

The domain model in Figure 1 reflects the complexity of the trust evaluation process. Payloads are seen to inherit trust from the Cybercraft platform and each association represents a possible trust decision that requires evaluation. There will have to be an initial trust value for a Cybercraft to trust the machine it is

loaded to. Computer networks work at the speed of light and are constantly changing and so as time goes on, trust evolution and delegation will need to be addressed. Thompson (Thompson 1984) demonstrated through a series of examples that it was impossible to trust any code unless personally written. Although for code to be secure it must be written by a trusted entity, a trust model can be used and applied to the Cybercraft domain to elaborate all trust relationships, as well as those not represented by a blue line in Figure 1.

2.3 Establishing a Root of Trust

We define a fundamental aspect of trust in Cybercraft as the ability for a system to behave as designed and intended. The notion of a "root of trust" based on hardware that cannot change is not new—and in fact has been a prized goal for organizations such as the Trusted Computing Group (TCG) for quite some time (Trusted Computing Group 2004). TCG specifications provide a starting point for an open set of security related building blocks that will associate trust with all aspects of computing to include storage, networking, software, mobile devices, personal computers, and servers. Two serious concerns for trusted computing standards such as those sponsored by TCG include the notion that competition may be stifled or that manufacturers may implement their "trusted" components incorrectly. In the context of Cybercraft, the former concern is not an issue as the military environment provides the operational bounds and the latter concern would need to be addressed adequately with any proposed Cybercraft platform solution. Nonetheless, the movement towards implementable and procurable secure hardware solutions in the commercial market provides perfect overlap with Cybercraft goals to integrate such technology.

We may liken trust establishment in hardware, software, or even the network itself to the establishment of trust with a service organization. TCG propose the use of silicon-based components such as the Trusted Platform Module (TPM) as a source of trusted storage where keys or passwords may be stored. At a minimum, we require a boot-time process to ensure secure configuration of all further system activity in order for the Cybercraft platform to establish the root of trust. We need to find trust models to capture this Cybercraft aspect and models which exercise further transitive relationships past the platform. Candidate trust models should also address the possibility of physical compromise (capture and subjection of hardware/software to adversarial activity) for either the platform or any possible payload. TCG already distinguishes different roots of trust including measurement, storage, and reporting, which find close corollary to proposed parts of the Cybercraft platform. In the trusted computed realm, we consider attestation as the processes for guaranteeing the accuracy of information and the ability of a platform to vouch for the trustworthiness of another platform. Attestation also provides a parallel notion for a major perceived computing paradigm supported by the Cybercraft architecture involving multi-agent cooperation between payloads accomplishing common tasks and goals.

The root of trust established in hardware for the Cybercraft platform gives us the basis for analyzing transitive trust decisions and gives us a framework to analyze possible trust models. In order to address how we may integrate these models into the development for Cybercraft, we begin with our approach for capturing functional and non-functional requirements, which we discuss next.

3. Cybercraft Requirements Distillation

One of the first steps in creating software is defining what it will be used for, in other words, requirements. Current practices for software development include the use of an iterative approach which assumes change and relies heavily on feedback. We apply the Unified Process (Larman 2005) as a model by which we can perform iterative analysis and design for the Cybercraft architecture while collectively identifying and refining requirements. While some aspects of the future Cybercraft vision may still be in the realm of research and development, the basic requirements for the system derive from a desire to provide comprehensive network defense services in a holistic and secure manner. The requirements for such a system are enormous to say the least and in order to know where to start, we begin with a general understanding of network defense missions as they are currently conducted. However, in order to capture the needs of a future system versus the closed context of current systems, we seek to define the network defense role from a general application perspective. To accomplish this task we use three distinct techniques in conjunction: attack/defense trees, use cases, and threat modeling.

3.1 Attack / Defense Trees

If we had free reign to design a holistic approach to network security, based on an extensible architecture that uses mission-specific co-operating payloads, we may best discover the possible tasks of the payloads by looking first at the possible ways our network is attacked. Attack trees provide a textual and visual means to analyze such attacks upon a system and are useful tools to reason about system security. Figure 2 shows an example attack tree where the root node is the attacker's goal and the children nodes show means or ways the attacker could accomplish an attack goal. The tree may have AND nodes which mean all child nodes must be successful to achieve the main goal. Trees use OR nodes to represent that only one of its child nodes need to succeed for its path to be successful (Edge 2007).

Once we exhaustively consider how our networks may be attacked and document those in the form of attack trees, we can then know best "what" a network defense architecture should be doing in response. For Cybercraft, we apply this approach iteratively by taking specific attacks and creating defense trees in response. Defense trees outline the possible mitigating actions we may take in response. The defense tree corresponds to the leaf nodes of the attack tree and provides us a "task" level understanding of what needs to be done. Figure 2 illustrates such trees as dotted lines and in this case shows that a DDoS attack may be defended against using firewall/switch/router ACLs or an intrusion detection system (IDS). Though very high level and general in this example, we envision such attack/defense tree modeling will provide a root level of understanding for possible Cybercraft tasks.

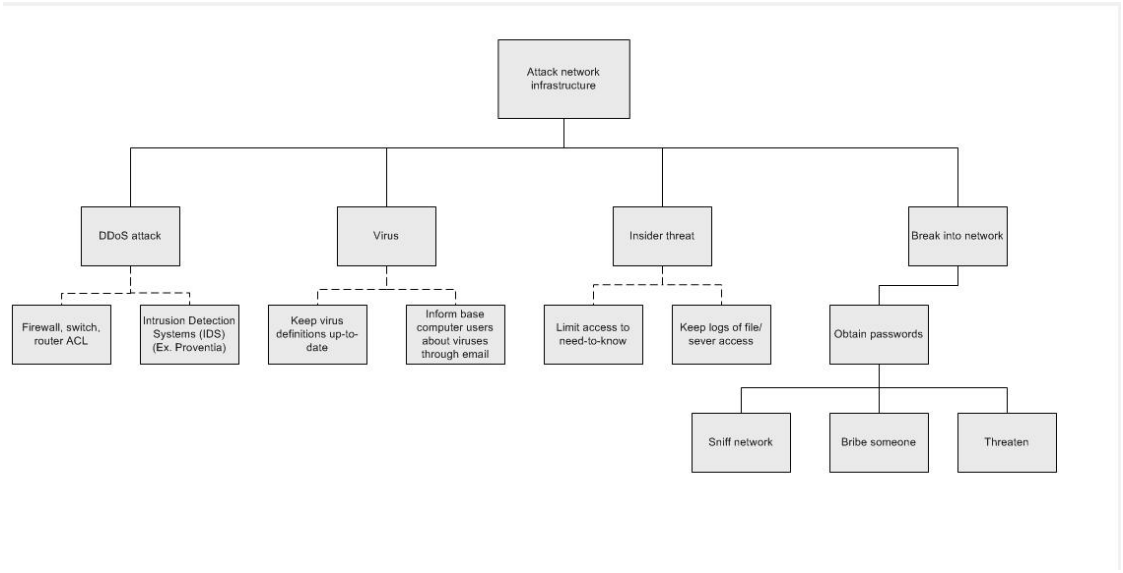

Figure 2: Attack tree created for an attack on network infrastructure

Once attack/defense trees are developed, we take some small starting number of trees/branches (our most important roles for example). We then analyze whether those defensive roles are currently being done by a human or an existing tool or both. In some cases, their may be no (effective) current method that mitigates, detects, or prevents certain attacks. This analysis method gives us a basis to determine whether we want the Cybercraft platform/payload architecture to do a particular task, do a current task better, or possibly automate an existing human-driven process. This process gives us the chance to not only analyze how well (or not) we currently do network defense, but also gives us the ability to look into future requirements without limitation of what currently is possible.

3.2 Use Cases

Once we determine defensive roles that are applicable to Cybercraft, we utilize a standard means for capturing software requirements for those roles: use cases. Use cases are textual means of describing the step-by-step interaction between a user and a system. In the case of Cybercraft, we expect that text

stories, diagrams, and models will help us determine not only functional requirements for payloads, but also non-functional requirements related to command and control, visualization, and policy development. Because use cases are software methodology agnostic, they provide an ideal means to communicate requirements between users, analysts, and designers. Use cases provide a concrete means to further determine the nature and number of payloads that support a given defensive role or mission. They also provide the ideal means to analyze the impact of transitive trust relationships in a concrete manner.

3.3 Threat Modeling / Misuse Cases

Not only do Cybercraft requirements need to address "how" we intend to perform network defense, they also need to uniformly address all possible avenues of vulnerability. We see the nature of the trust question most clearly in this context as it forces us to consider possible ways in which the actual Cybercraft may itself be subverted. Threat modeling allows a team to identify the highest risk components in an application (Swiderski 2004) and possible areas of vulnerability within software architecture. Organizations like Microsoft uses threat modeling as an extra means to find threats in their software applications. Effective threat modeling should begin early in the software design lifecycle and begin to reason about possible security flaws.

We use scenarios, dependencies, and any implementation assumptions as part of the analysis process to help identify trust expressions (or questions) that should be evaluated or answered. To aid in this process, we also apply the art of "misuse cases" to the normal use case process. Misuse cases are simply step-by-step descriptions that detail adversarial action and records how the system (should or should not) respond accordingly. We expect this analysis to directly feed our requirements for trust model expression and exercise, which we discuss in the next section.

4. Trust Model Evaluation

Given an initial set of use cases, misuse cases, and attack trees and given a robust approach to fully flush out use case details in an iterative approach, we feel confident that we have a workable approach to capture requirements for the Cybercraft system. This approach may be useful for other network-based defensive systems as a means to consolidate requirements and build the system in an iterative process. In order to address the issue of self protection and trust, we consider the unique aspects of the Cybercraft architecture that need trust model expression and that are revealed as part of the requirements analysis process. We consider several models such as hTrust, VTrust, SECURE, MARISM-A, SATEMA, and I-TRUST. hTrust (Capra 2004), mimics the interactions of humans trust and works well in mobile settings because of the minimal resource demands. VTtrust (Ray 2004 and Ray 2005), is a vector-based trust model. Trust interactions are represented as relational entities translated to a central database. SECURE is very similar to hTrust and tries to mimic human trust as well (Cahill 2003). MARISM-A (Robles 2002) is an architecture for mobile agents with recursive itinerary and secure migration. SATEMA (Foster 2005 and Varadharajan 2000) is an architecture for application to secure agent internet based applications and I-TRUST (Tang 2003) is a trust model that captures the trust between agent and user.

For each trust model, there are three ideas of trust: initial trust, trust exchange, and trust evolution. Initial trust is the first formation of trust between two agents. Trust exchange deals with the protocols and exchange of trust between agents. Trust evolution is the continuation of trust over time. This allows for the decay of knowledge that happens over time. Each trust model uses various words for each of these trust ideas but essentially mean the same thing. Table I shows each model with their terms.

Table 1: A summary of similar trust ideas for each trust model

General Trust Ideas from the Trust Models						
	hTrust	VTrust	SECURE	MARISM-A	SATEMA	I-TRUST
Initial trust	formation	knowledge	formation	N/A	N/A	initial
Trust exchange	dissemination	experience	evolution	N/A	N/A	continuing trust
Trust evolution	evolution	recommended	recommended	N/A	N/A	recommended

4.1 hTrust

hTrust (Capra 2004) is made up of three main parts: trust formation, trust dissemination, and trust evolution. Trust formation is the initial trust before an interaction occurs; creating a trusting environment that gives us a prediction of trustworthiness. Trust dissemination uses a recommendation exchange protocol to exchange trust opinions. The evolution of trust is the part that allows for a continuous self-adaptation of trust.

4.2 VTrust

The trust relationship in VTrust (Ray 2004 and Ray 2005) consists of a vector with three components: experience, knowledge, and recommendation. Experience is the number of events two agents share within a certain timeframe. Knowledge is composed of direct knowledge and indirect knowledge. A recommendation uses a recommendation value as its basis for trust. Stevens considers the application of trust vectors and their applicability for Cybercraft fitness (Stevens and Williams 2007). Their analysis examines the use of cybercraft payloads in multi-agent information gathering roles where agents may have to evaluate information from other agents. Such roles cover a large number of defense applications where network sensing data are analyzed. Their conclusions show that a modified Trust Vector model could meet the needs for expressing trust decay and transitive trust decisions in the information retrieval context.

4.3 SECURE

The Secure Environments for Collaboration among Ubiquitous Roaming Entities (SECURE) (Cahill 2003) project uses the same ideas as hTrust with human trust interactions as the basis of the model. This model allows for decisions to be made based on partial information and the ability to reason about trust and risk. The three ideas about trust for this model are: trust formation, trust evolution, and recommendations. In forming trust, the risks are decomposed by possible outcomes. Risk analysis is performed for each outcome and a cost-PDF (probability density function) is created to represent the distribution of costs. The risk analysis weighs the evidence on whether a principle is trustworthy against the risks if it isn't.

Building or gaining trust happens through personal observations of previous interactions and recommendations. Personal experience between two entities influence trust more than recommendations from partially trusted third-parties and are essential for the subjective aspect of evaluating trustworthiness. Recommendations circulate trust by providing supporting evidence for trust decisions in unknown entities.

4.4 SATEMA

The security and trust enhanced mobile agent (SATEMA) (Foster 2005 and Varadharajan 2000) architecture uses trust as part of the decision making process. The architecture has a trusted authority, a clearing house, which is a central repository for data and all the trust values between the different entities. Trust levels are obtained from the clearing house for decision making.

4.5 MARISM-A

The MARISM-A (Robles 2002) architecture implements secure migration, secure agent communication and a coexistence of different agent architectures. The migration protocol transfers agents and uses SSL beneath the protocol. Secure communication is implemented by an underlying PKI and different agent architectures were created within MARISM-A for specific security mechanisms and requirements.

4.6 I-TRUST

I-TRUST (Tang 2003) investigates trust between user and agent in a multi-agent portfolio management system. User models are created to form the initial trust between user and agent. Continuing trust is held by agents using a learning technique to adapt to the user and maximizing the reward from the

environment. Recommendations come in the form of agent suggestions and whether or not the expected outcome is achieved or not.

Trust expression in the Cybercraft domain centers around several key relationships: platform to payload, platform to platform, platform to node, and payload to payload. There are obviously many others to consider. Our continuing work considers which, if any, of the current trust models best express proof that trust transfers from the root of trust in the payload to other components in the system.

5. Conclusion

There are many unknowns still to be captured and analyzed for Cybercraft. We pose in this paper a unique approach to consider network-defense requirements for this future application environment by using software engineering-based use cases, threat modeling, and attack/defense tree analysis. Based on the overarching question of whether future commanders may rightly place trust in the Cybercraft architecture to defend friendly networks, we show how our requirements process supports the development of trust-specific parameters which may in turn be evaluated in a candidate trust model. We briefly present issues and needs for trust models which allow us to express and extend the root of trust which the Cybercraft platform must ensure. Our future work includes iterative requirements exposition for Cybercraft and enumeration of the full trust relationships within the Cybercraft domain- with the ultimate goal of evaluating trust model efficacy.

References

Bishop, M. (2003) *Computer Security: Art and Science,* Addison Wesley.

Cahill, V., Shand, B., English, C., Serugendo, G., Carbone, M. (2003) "Using Trust for Secure Collaboration in Uncertain Environments", *IEEE Pervasive Computing*, vol. 2, no. 3, p. 52.

Capra, L. (2004) "Engineering human trust in mobile system collaborations," *SIGSOFT '04/FSE-12: Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*, New York, NY, USA, pp. 107–116, ACM Press.

DOD. (2006) "Air Force ACC IO RAWG 2006"

Edge, K. S. (2007) "A Framework for Analyzing and Mitigating the Vulnerabilities of Complex Systems via Attack and Protection Trees", PhD thesis, Air Force Institute of Technology.

Foster, D. and Varadharajan V. (2005) "Security and Trust Enhanced Mobile Agent Based System Design", *Information Technology and Applications, 2005, ICITA 2005, Third International Conference*, vol. 1, pp. 155-160.

Gambetta, D. (2000) "Can We Trust Trust?", *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, Oxford, pp. 213-237.

Gettle, M. (2005) "Air force releases new mision statement", [online], http://www.af.mil/news/story.asp?id=123013440.

Glass, R. L. (2003) *Facts and Fallacies of Software Engineering,* Addison Wesley.

Gollmann, D. (2005) "Why trust is bad for security", *Proceedings of the First International Workshop on Security and Trust Management (STM 2005), ENTCS 157:3*, 25 May 2006, pp. 3-9.

Grandison, T. and Sloman, M. (2000) "A Survey of Trust in Internet Applications", *IEEE Communications Surveys and Tutorials,* vol 3.

Howard, M., and Whittaker, J. A. (2005) "Demystifying the Threat-Modeling Process", *Security and Privacy Magazine, IEEE*, vol. 3, no. 5, pp. 66-70.

I. D. Air Force Research Laboratory. (2007) *Conference slides from Cyber Defense Conference*.

Larman, C. (2005) *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development,* Prentice Hall PTR, 3 ed.

Lopez, C. T. (2006) "8th air force to become new cyber command", [online], http://www.af.mil/news/story.asp?storyID=123030505.

Phister, D. P. J., Fayette, D., and Krzysiak, E. (2006) "Cybercraft: Concept linking NCW Principles with the Cyber Domain in an Urban Operational Environment".

Ray, I. and Chakraborty S. (2004) “A vector model of trust for developing trustworthy systems”, *Proceedings of 9th European Symposium on Research in Computer Security* (ESORICS’04), Sophia Antipolis, France.

Ray, I. and Chakraborty S. (2005) “Vtrust: A trust management system based on a vector model of trust,” *Proceedings of 1st International Conference on Information Systems Security (ICISS’05)*, Sophia Antipolis, France.

Robles, S., Mir, J., and Borrell, J. (2002) “Marism-a: An architecture for Mobile Agents with Recursive Itinerary and Secure Migration”, *2nd IW on Security of Mobile Multiagent Systems*.

Sailer, R., Van Doorn L., Ward, J. (2004) “The Role of TPM in Enterprise Security”, IBM Research Report RC23363.

Swiderski, F. and Snyder W. (2004) *Threat Modeling*. Microsoft Press.

Stevens, M. and Williams, P.D. (2007) “Use of Trust Vectors for CyberCraft and the Limits of Usable Data History for Trust Vectors”, *IEEE Symposium on Computational Intelligence in Security and Defense Applications*, pp. 193-200.

Tang, T., Winoto, P., and Niu, X. (2003) “I-TRUST: Investigating Trust Between Users and Agents in a Multi-agent Portfolio Management System”, *Electronic Commerce Research And Applications*, vol. 2, no. 4, pp. 302-314.

Thompson, K. (1984) “Reflections on trusting trust,” *Communications of the ACM*, vol. 27, no. 8, pp. 761–763.

Trusted Computing Group, TCG Specification Architecture Overview. Revision 1.4 2 Apr 2007. URL: https://www.trustedcomputinggroup.org/groups/TCG_1_4_Architecture_Overview.pdf.

Varadharajan, V. (2000) “Security Enhanced Mobile Agents”, *Proceedings of the ACM Conference on Computer and Communications Security*