PNNL-19117

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Trust Management Considerations for the Cooperative Infrastructure Defense Framework: Trust Relationships, Evidence, and Decisions

WM Maiden

December 2009

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle Since 1965

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

**Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm**

Abstract

Cooperative Infrastructure Defense (CID) is a hierarchical, agent-based, adaptive, cyber-security framework designed to collaboratively protect multiple enclaves or organizations participating in a complex infrastructure. CID employs a swarm of lightweight, mobile agents called Sensors designed to roam hosts throughout a security enclave to find indications of anomalies and report them to host-based Sentinels. The Sensors' findings become pieces of a larger puzzle, which the Sentinel puts together to determine the problem and respond per policy as given by the enclave-level Sergeant agent. Horizontally across multiple enclaves and vertically within each enclave, authentication and access control technologies are necessary but insufficient authorization mechanisms to ensure that CID agents continue to fulfill their roles in a trustworthy manner. Trust management fills the gap, providing mechanisms to detect malicious agents and offering more robust mechanisms for authorization. This paper identifies the trust relationships throughout the CID hierarchy, the types of trust evidence that could be gathered, and the actions that the CID system could take if an entity is determined to be untrustworthy.

Notice

This document pertains to CID as designed and implemented through December 2008 [11].
Additional trust management work pertaining to CID was previously published in [1].

Acknowledgements

The author would like to gratefully acknowledge the contributions of reviewers Glenn Fink (CID principal investigator), Jereme Haack, and A. David McKinnon.

Table of Contents

Introduction

The Cooperative Infrastructure Defense (CID) framework uses a hierarchy of rational agents (see Figure 1) to monitor and respond to cyber security issues on hosts within and across the security enclaves that constitute a complex infrastructure such as the electric power grid. Because humans are ultimately responsible for the actions of their cyber defense systems, CID's hierarchy includes human Supervisors at the top of the hierarchy where they receive situational awareness from their agents and provide policy-based direction to the agents.

At the lowest level of the hierarchy, CID employs a swarm of lightweight mobile agents called Sensors, each with a narrowly-targeted classifier. Modeled after social insects, the Sensors randomly roam throughout a security enclave to detect specific signatures and/or to compare each host they visit with previously-visited hosts to detect anomalies based on discovered patterns or ranges believed to be normal. When a Sensor reports its findings to the host-based Sentinel agent, the Sentinel agent combines the information from the Sensor with information from other Sensors and its own knowledge of the host to determine the usefulness of the new information provided by the Sensor. If the information is deemed useful, the Sentinel "rewards" the Sensor with additional energy. The rewarded Sensor is then able to drop digital pheromone on a neighboring host as it leaves. The pheromone attracts the attention of other Sensors to the vicinity; these Sensors will apply their own distinct classifiers to detect and report additional problem indicators.

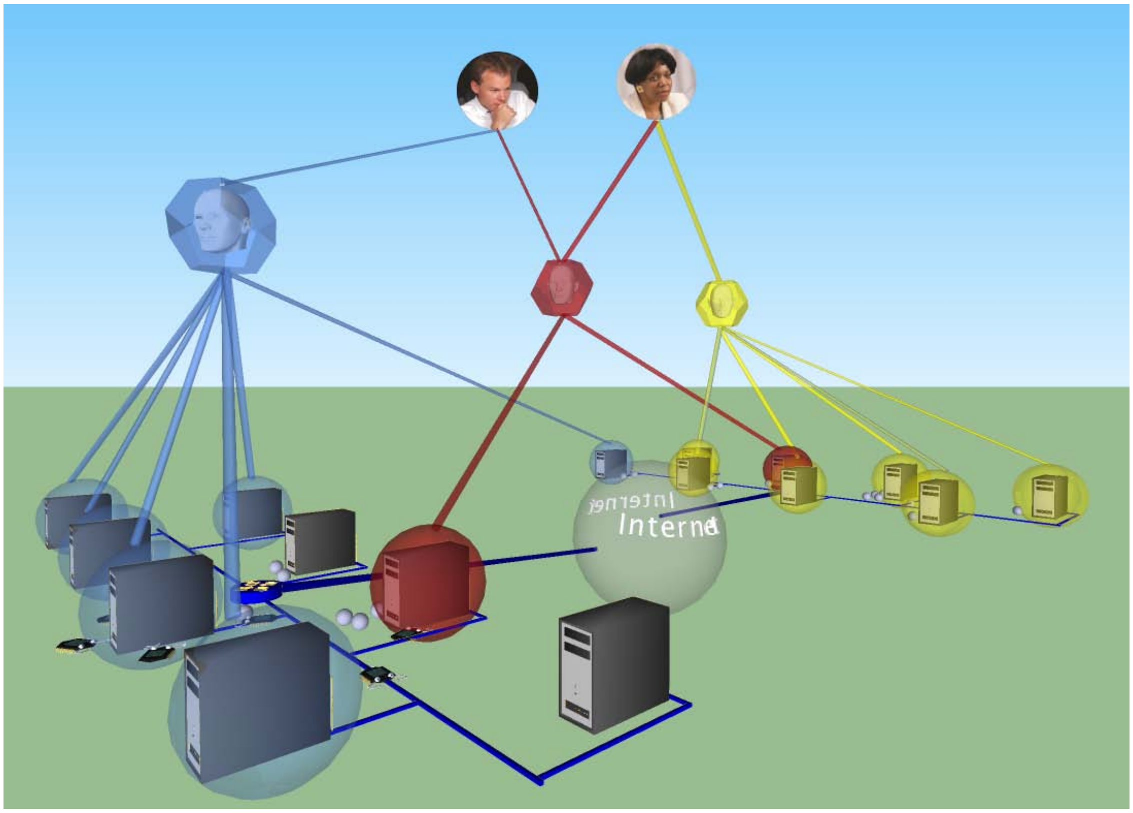


Figure 1. The CID Hierarchy. Top-to-bottom: Supervisors, Sergeants, Sentinels (resident on hosts), and Sensors (depicted as digital ants).

The Sensors' findings are like pieces of a larger puzzle, which the Sentinel puts together to determine the problem. The Sentinel uses reinforcement learning in combination with supervised learning to evaluate the findings. When it diagnoses a problem, the Sentinel fixes it according to policy and reports the problem and resolution to the enclave-level Sergeant agent crediting the Sensors whose findings resulted in successful diagnosis.

The Sergeant, which is responsible for overall enclave security, dialogues with the human Supervisor about high-level policy guidance for the enclave. From this, the Sergeant creates a set of executable policy statements which it passes, with the appropriate delegations, to the Sentinels to implement on the host(s) they manage. The Sergeant also tracks which Sensor combinations are useful for detecting which problems and makes the classifier code for those combinations available to its peers—the Sergeants of other security enclaves. The Sergeant may also be authorized by the Supervisor to make service-level agreements with other Sergeants.

Although CID's hierarchical interactions occur within a single security enclave, its agents should not be blindly trusted, nor should authentication and role-based access be considered sufficient authorization. Because it is a security system, its agents could be targets of attack by malicious entities inside and outside of the security enclave. The Sensors' mobility increases their exposure to potential malicious forces, as does the fact that they must not avoid going to hosts that are exhibiting problems. Sentinels are at risk if the host on which they reside is attacked by a malicious entity. Trust becomes especially complex when a Sentinel or Sergeant is compromised and begins to act maliciously. Trust management can be used to alleviate these potential weaknesses.

Section 1 provides an overview of trust management. Subsequent sections look at trust relationships throughout the CID hierarchy, potential sources of trust evidence for each type of CID agent, and potential actions that could be taken against each type of CID agent in response to low trust ratings.

1.0 Trust Management Overview

Trust has many different interpretations depending on the context. It can pertain to authentication, authorization, competence, reliability, integrity, dependability, timeliness, accuracy, or any combination of these properties. Authentication and authorization are the "hard" side of trust; they are determined by the policies and credentials of a structured environment. The remaining attributes are the "soft", and often social, side of trust. They speak to quality of service (QoS) and are not black-and-white, but rather measured in degrees and changeable over time as one's perception of an entity's reputation is formed through direct personal experience and/or through the recommendations of others based on their experience.

1.1 Trust Context Classification

Trust is context-specific; the trust that entity A (the *trustor*) has in entity B (the *trustee*) will vary depending on the specific context. The types of context that can occur in a distributed system can be classified as follows:

- **The trustor must decide whether to grant a trustee access to a resource.** In this context there may be a policy decision point (PDP) that considers the trustee's trust evidence. The trust evidence may include its own authentication or authorization credentials, credentials delegated by others, and/or its reputation as known by the trustor and other entities.

- **The trustor wants to select a trustee that will provide a quality service.** In this context, if the trustor has several service providers to choose from, the selection may be based on each service's quality of service (QoS) to other trustors as well as any previous direct experience the trustor has in using the trustee's service.

- **Infrastructure trust** pertains to the foundational trust that an entity must be able to have in the hardware, operating system components, and networks that form the infrastructure upon which the trust relationship takes place.

1.2 Foundations for Trust

Wilhelm et al identified four foundations for trust [3]. Although their paper was specific to trust in mobile agents, the four foundations for trust are broadly applicable:

- **Blind trust**, such as the trust a child has for a parent.

- **Trust based on control and punishment**, such as the threat of loss of employment, fines, or jail time. This is typically applicable in situations where there is a controlling entity such as a government or employer. It is more applicable to trust in humans than trust in software entities.

- **Trust based on a good reputation**. This type of trust assumes that because a good reputation is hard to build and easily destroyed, the entity will not want to do anything to harm its reputation. One or more trust *evidences* are used to form the basis of an entity's reputation.

- **Trust based on policy enforcement**. This is trust in the policy and the enforcement mechanism rather than in the entity itself. It includes the enforcement provided by standard security mechanisms such as credentials.

1.3 Trust Management Defined

The term "trust management" was coined by Matt Blaze of AT&T Research Labs, in 1996 [4]. Blaze's concept of trust management centered on specifying security policies and applying them to authorization statements embedded in credentials to enable a trust management engine to directly assess whether a requested action should be allowed. Blaze's use of the term reflects *trust based on policy enforcement*. More recently Tyrone Grandison [5] defined trust more generally to include both *trust based on policy enforcement* and *trust based on a good reputation*. He defines trust management as "the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships."

Trust management systems are generally based on reputation evidence [6] [9], credential-based evidence supporting policy enforcement [4] [7] or both [5] [8]. Trust management is most commonly used in distributed systems where there is no single authority, such as an employer or civil authority, that can control and levy punishment for inappropriate actions, and for which blind trust is not a viable option (e.g., because of the probability of a trust violation coupled with the associated risks).

Reputation-based trust management mechanisms are common in e-commerce [2] where a consumer needs to minimize the risk involved in using a service (e.g., purchasing a product) from an unfamiliar provider. Reputation is also useful in electronic communities, such as peer-to-peer [9] or wireless sensor networks [10] to detect nodes that are not being good citizens of the community, such as entities that freely consume resources without offering any resources in return. The risk to be avoided may be maliciousness or simply poor QoS. Trust management was originally conceived as a method for establishing trust across security enclaves where authentication is either impossible or meaningless, but reputation-based trust is also useful for maintaining trust within a security enclave when insider threat, intruders, or deteriorating QoS is a concern.

Credential-based trust management systems typically have to do with authorization or delegation. In credential-based trust management systems, formally-specified policy statements are used by service providers or resource owners in conjunction with authentication certificates and/or authorization credentials to determine if consumers have the right to use a service or resource. This process may occur through a trust negotiation process wherein each entity iteratively reveals either policy statements or credentials to the other until trust has been established. Although authentication certificates, such as X.509 certificates, are useful within a security enclave to prove identity, attribute credentials and peer-granted identity certificates (e.g. PGP) can be used to control access to resources in distributed systems that cross security enclaves where there are no shared authentication certificates controlled by a central authority. Even when a central authority spans enclaves, authentication merely serves to prove identity, not trustworthiness.

Credentials also provide much more expressiveness than the typical read/write/execute access control permissions. For example, an authorization credential might represent that “the holder is authorized to sign contracts worth up to $200,000” or a policy may require that only university students are eligible to sign up for a benefit and therefore a “student” attribute credential signed by a known university must be supplied.

Both forms of trust management have the potential to provide value to CID.

2.0 Trust Relationships in CID

2.1 General Patterns and Observations

Although trust relationships vary based on context, the general pattern of CID trust relationships is shown in Table 1. Proof of identity is checked in most cases in addition to the trust foundation listed in the cells of the table. The next section provides a more detailed itemization of the context-specific trust relationships in CID with a brief assessment of the foundations on which trust can be built for each relationship.

Table 1. Overview of Direct Trust Relationships in CID

	Trusted Entity			
Trusting Entity	**Sensor**	**Sentinel**	**Sergeant**	**Supervisor**
Sensor	Indirectly (source of pheromone is checked by Sentinel that receives the pheromone)	Reputation	NA	NA
Sentinel	Reputation of sending Sentinel + Policy Enforcement	Reputation	Blind Trust	NA
Sergeant	NA	Reputation	Credentials + Reputation	Blind Trust
Supervisor	NA	NA	Blind Trust	NA

The following general observations can be made:

- Supervisors, Sergeants, and Sentinels should have public/private keys that can be used to authenticate them, to conclusively confirm them as the source of a message, and to log their actions for non-repudiation purposes.

- The Supervisor and Sergeant are blindly trusted within their enclave once they have been authenticated, although digitally-signed logging for non-repudiation is also recommended.

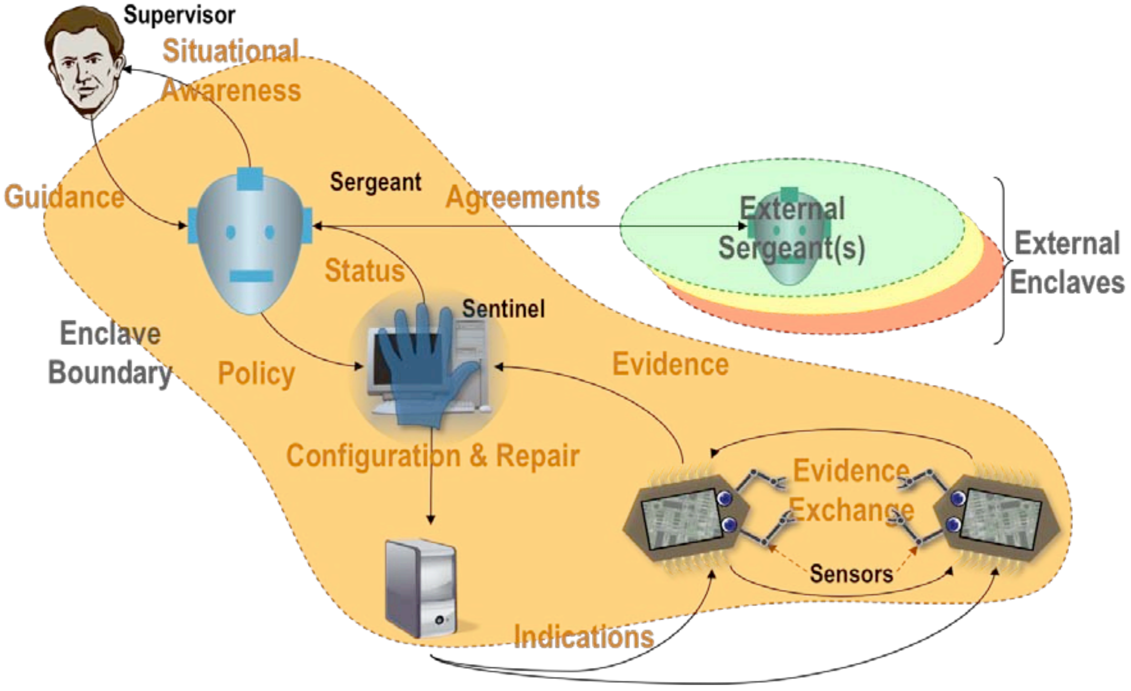
- The Sentinel is subject to reputation-checking due to its location vulnerability. Sensors are trusted so long as the Sentinel that created them and the Sentinels that send and receive them are trusted.

- Sensor trust is established indirectly since they are too numerous and short-lived.

2.2 Detailed Analysis

In the remainder of this section, the CID trust relationships, many of which are shown graphically in Figure 2, are detailed as belonging to one of the three types of trust contexts – trust to access resources, trust in a service, and infrastructure trust. In each case the potential foundations for trust are noted.

Figure 2. CID Relationships



2.2.1 Trust to Grant Access to a Resource

This type of trust, *trust to grant access to resources*, is from the perspective of the resource provider who needs a policy decision point (PDP) to protect a resource or service from unauthorized users.

In most CID cases (in contrast to the typical resource granting scenario), the resource is pushed to the recipient or pre-allocated to the recipient rather than requested by the recipient. This provides a stronger degree of control from the start since the entity that is pushing the information must already know the receiving entity (or list of entities) to whom the information must be sent; the recipients are not unknown identities.

Table 2 lists CID resource providers, the resources they need to protect or control, the consumer/user, and the potential trust foundation – blind trust, control and punishment, reputation, or policy enforcement. The latter column also indicates any standard security measures that could be used in place of or in addition to trust management, so accurate decisions can be made with regard to where trust management should be applied in CID.

All CID entities within the context of a single security enclave are subject to authentication using identity certificates issued by a trusted third party. The exception is the cross-enclave Sergeant/Sergeant relationship.

Table 2. CID relationships pertaining to protection and control of resources

Resource to be Protected / Controlled	**Resource Controller** (trustor)	**Consumer / User** (trustee)	**Pushed / Requested**	**Potential Foundation for Trust**
Policy dialog	Supervisor	Sergeant	Pushed	Blind Trust

Resource to be Protected / Controlled	**Resource Controller** (trustor)	**Consumer / User** (trustee)	**Pushed / Requested**	**Potential Foundation for Trust**
			(initiated by Super-visor) / Requested (clarification requested by Sergeant)	Policy enforcement: • Verify that role is Sergeant and parent is the Supervisor • Dialog (or decisions) must be digitally signed and optionally logged by the Sergeant
Geography (the set of hosts in the enclave to which Sentinels can allow Sensors to move)	Sergeant	Sentinels	Pushed	Policy enforcement: • Sergeant maintains a current list of authorized Sentinels • Verify that role is Sentinel and parent is the Sergeant • The geography update must be logged by both the Sentinel and the Sergeant, and each log entry must be digitally signed Reputation: • Check the Sentinel's reputation before sending the geography update. (Optional. The geography may not be sufficiently sensitive to warrant the overhead of checking the reputation.)
Policy statements	Sergeant	Sentinels	Pushed	Policy enforcement: • Verify that role is Sentinel and parent is the Sergeant • The policy statements must be logged by both the Sentinel and the Sergeant, and each log entry must be digitally signed
Execute permission, and ability to modify system configuration	Host (system administrator)	Sentinel	Pre-allocated	Policy enforcement: • Privileges are granted to the Sentinel upon installation. • Confirm that Sentinel's

Resource to be Protected / Controlled	**Resource Controller** (trustor)	**Consumer / User** (trustee)	**Pushed / Requested**	**Potential Foundation for Trust**
				key is signed by the Sergeant. By granting these privileges to the Sentinel, the Sentinel is made responsible for establishing a PDP for controlling access to the host by the Sensors. Therefore, CID treats the Sentinel as a proxy for the host.
Share of limited CPU, memory, and disk resources	Host (system administrator)	Sentinel	Pre-allocated where possible, else requested	Blind Trust (unless allocations can be restricted upon installation) Control and punishment: • Resource monitoring (optional)
Execute permission and read access to system logs; share of limited CPU, memory, and disk resources	Receiving Sentinel	Sensors	Requested	Policy enforcement: • Limit permissions to read and execute; no writing • Sandboxing • Limit amount of resources dedicated to a Sensor • Limit number of Sensors allowed on the platform (log this number; digitally signed) • Authenticate the sending Sentinel prior to accepting the Sensor or allocating resources to it. • Static verification of Sensor code (via digitally signed hash) upon arrival. • Log each Sensor received and the sending Sentinel. Reputation: • Check sending Sentinel's reputation • Check creating Sentinel's reputation • Decrement the sending

Resource to be Protected / Controlled	**Resource Controller** (trustor)	**Consumer / User** (trustee)	**Pushed / Requested**	**Potential Foundation for Trust**
				Sentinel's reputation if the Sensor causes problems on the host.
Sensor data	Sensor	Receiving Sentinel		Reputation: • Sensor (or sending Sentinel) checks receiving Sentinel's trust level prior to moving.

2.2.2 Trust in a Service

Trust in a service represents the perspective of the resource consumer who needs to be able to trust services provided by another entity. This includes entities higher in the CID hierarchy which must be able to trust the entities under them to perform their duties.

Table 3. CID Relationships pertaining to trust in a service

Consumer / User (trustor)	**Service to be Used**	**Service Provider** (trustee)	**Potential Foundation for Trust**
Supervisor	Situational awareness	Sergeant	Blind Trust: • Implicitly trust Sergeant, but observe the process (e.g., look for a hung process). Accuracy and timeliness issues may be the result of inefficiency in the code or with host or network throughput rather than maliciousness. Policy Enforcement: • Verify that role is Sergeant and parent is the Supervisor • Sergeant must log (digitally signed) the situational awareness reports that exceed a given importance threshold.
Supervisor	Interpret and enforce policy	Sergeant	Blind Trust: • Implicitly trust the Sergeant, but monitor actions and results. Policy Enforcement: • Verify that role is Sergeant and parent is the Supervisor Control and punishment:

Consumer / User (trustor)	**Service to be Used**	**Service Provider** (trustee)	**Potential Foundation for Trust**
			- The Sergeant is programmed to modify its behavior to maximize the value of rewards received from the Supervisor.
Supervisor	Authorization to negotiate with other Sergeants	Sergeant	Trust via policy enforcement: - An authorization credential signed by the Supervisor can be given to the Sergeant to prove its authorization to peers. The Supervisor demonstrates degrees of trust in the Sergeant by granting credentials containing levels of authorization.
Sergeant	Policy guidance	Supervisor	Blind Trust Policy Enforcement: - Verify that role is Supervisor and that Supervisor's key matches the Sergeant's parent's key Policy Enforcement: - Log all policy changes and include the timestamp and identification of the Supervisor that made the policy change. The Supervisor's private key should be used to sign the log for non-repudiation. Although non-repudiation isn't usually discussed in trust management literature as a foundation for trust, it does serve this purpose through control and punishment.
Sergeant	Sensor logic or service agreements offered by a Sergeant from another enclave	Other Sergeants	Policy Enforcement: - Credential-based trust negotiation Reputation: - Reputation is used in peer-to-peer systems to detect when members are providing something bad or are just not "pulling their own weight" in the community.
Sergeant	Implement policy	Sentinel	Policy Enforcement: - Verify that role is Sentinel and parent is the Sergeant Reputation: - Where possible, independently verify policy implementation and use this as input to a Sentinel's reputation. Consider using a Sensor to compare logs and settings of Sentinels vs. the Sergeant's version. Would need to be rewarded by Sergeant instead.
Sergeant	Accurate,	Supervisor	Blind Trust

Consumer / User (trustor)	**Service to be Used**	**Service Provider** (trustee)	**Potential Foundation for Trust**
	actionable, and responsible policy dialog		Policy enforcement: • Verify that role is Supervisor and that Supervisor's key matches the Sergeant's parent's key
Sergeant	Accurate and timely status	Sentinel	Policy Enforcement: • Log time of request and time of receipt of information from the Sentinel. Reputation: • If accuracy or timeliness suffers, downgrade the Sentinel's reputation.
Sentinel	Geography (the set of hosts in the enclave to which Sentinels can allow Sensors to move)	Sergeant	Blind Trust Policy enforcement: • Verify that role is Sergeant and that Sergeant's key matches the Sentinel's parent's key • Geography received by Sentinel must be digitally signed by the Sergeant and logged by both the Sentinel and the Sergeant.
Sentinel	Accurate and actionable policy	Sergeant	Blind Trust Policy enforcement: • Verify that role is Sergeant and that Sergeant's key matches the Sentinel's parent's key • Sergeant and Sentinel should both log (and digitally sign) all policy changes and include the timestamp
Sentinel	Accurate and timely information on what the Sensor found on the Sentinel's Host	Sensors	Perform checks on and before arrival (as described in Table 2), then Blind Trust.
Sentinel	Provide pheromone	Sensors	Policy Enforcement: • Before accepting pheromone, the Sentinel should verify that the Sensor has a Sensor role credential with a chain leading back to the Sergeant. • Check the digitally signed hash of the Sensor's code (generated by the Sensor's creator) that the Sensor carries with it.
Host	Monitor Sentinel	Sergeant	Blind Trust

Consumer / User (trustor)	**Service to be Used**	**Service Provider** (trustee)	**Potential Foundation for Trust**
Host	Reasonable and timely resolution of problems found on the host	Sentinel	Indirect. The Host will implicitly trust the Sergeant to monitor the Sentinel.
Host	Monitor Sensors	Sentinel	Blind Trust or Host could have process to check neighbor's view of Sentinel reputation.
Host	Accurate and timely identification of problems	Sensors	Indirect. The Host will implicitly trust the Sentinel to monitor the Sensors.
Sensor	Provide reward when the Sensor has detected and reported on a problem	Sentinel	Reputation: • Sensor (or sending Sentinel) checks receiving Sentinel's trust level prior to moving
Sensor	Routing to neighboring hosts	Sentinel	Reputation: • Sensor (or sending Sentinel) checks receiving Sentinel's trust level prior to moving
Sensor	Accurate and timely indication of a path toward a host of interest (i.e., digital pheromone)	Other Sensors	Blind Trust Policy enforcement: • Indirect through Sentinel

2.2.3 Infrastructure Trust

Infrastructure trust pertains to the systems and networks upon which delivery of the service depends. CID will blindly trust the networks, and is itself the mechanism for establishing host trust.

3.0 CID Trust Relationships Most Likely to Benefit from Trust Management

Many of CID's trust relationships listed in the previous section can be handled efficiently and effectively through traditional security mechanisms such as authentication, digital signatures, and logging. The following relationships, however, would benefit from the addition of trust management techniques.

3.1 Reputation-Based Trust Management

Reputation-based trust management using a distributed trust model has been successfully used in communities of peers such as P2P systems, wireless sensor networks, and multi-agent communities to detect when members are providing malicious feedback, bad data, or are just not "pulling their own weight" in the community. The Sergeant-to-Sergeant cross-enclave relationship is a community of peers. Reputation would provide a mechanism for ensuring that Sergeants will be detected and isolated if they pass bad or even malicious Sensor logic to other Sergeants or if they take advantage of others' experiences by using their shared Sensors without ever sharing their own useful Sensors for the good of the community.

Since Sentinel's reside on the hosts they monitor, they are vulnerable to corruption. To detect such corruption, the Sentinels' reputation should be monitored. The risk to Sentinels can also be reduced by having them operate on a Trusted Computing platform [12] or by moving them to a separate platform from which they monitor their assigned host(s).

Sensors are vulnerable because of their exposure to multiple hosts and because they must visit potentially-infected hosts. However, because of their quantity, brief lifetimes, and minimal interactions, reputation-based trust management is *not* recommended for Sensors. Instead, the trustworthiness of the Sensors' creator and sending Sentinel should be checked instead. If the creator is trustworthy then the Sensor was likely created trustworthy. If the sending Sentinel was trustworthy, then it is unlikely that the Sensor was corrupted while on the Sentinel's host. Issues with a Sensor should reflect on the reputation of the previous Sentinel and/or the creating Sentinel depending on whether the Sensor was modified since creation.

3.2 Policy-Based Trust Management

All of CID's delegation relationships require the definition of policy and creation and management of authorization credentials. Standard X.509 certificates could be used, but authorization credentials that specify finer-grained controls and cross-enclave credential negotiation would be especially useful for Sergeant-to-Sergeant relationships.

4.0 Reputation Trust Evidence in CID

The following evidences can be used to make the decision to lower the entity's reputation. The opposite evidence would raise its reputation.

In some cases below, the mechanism for detecting the trust evidence has not yet been determined. One possible scenario for some of these cases is that monitoring sensors could be created, dispatched, and rewarded by the Sergeant.

4.1 Sensor Trust Evidence

Negative Sensor trust evidence should impact the reputation of the creating Sentinel (if the Sensor has not changed since it was created) or impact the reputation of the preceding Sentinel (if the Sensor has changed). In addition, each of these trust evidences warrants the discontinuation of the Sensor's travels, either by killing the Sensor or forwarding it to the Sergeant for analysis along with an explanation of the violation.

- Privileges or resource consumption exceed the allowed configuration.
- The cryptographic hash of the serialized code is not the expected hash value.
    - This should be a security verification check prior to execution, in addition to impacting the appropriate Sentinel's reputation.
- The size of the arriving agent's memory is too large (carrying more info than it should be carrying)
    - Potential issue: Will we know what memory size is expected?
    - This should be a security verification check prior to execution, in addition to impacting the appropriate Sentinel's reputation.
- False reporting of data the Sentinel knows is not true.
- Sensor disrupts the host in some way (Specifics to be determined.)

Positive trust evidence should be generated for the creating Sentinel in proportion to the reward that the hosting Sentinel gives the Sensor.

4.2 Sentinel Trust Evidence

- Revoked or false credentials
- Policy or geography change logged by Sentinel does not match the Sergeant's log
- Policy or geography in use by the Sentinel does not match log
- Sentinel's privileges or resource consumption exceed the initial configuration.
- Discontinued or false reporting to the Sergeant
- Red Team sensor, created by the Sergeant, visits the Sentinel and the Sentinel fails to report it.
- Sensor reports to the next Sentinel the time lag from Sensor arrival to when the Sensor was allowed to run by the previous Sentinel.
- Killing an excessive number of Sensors

4.3 Sergeant Trust Evidence

- Revoked or false credentials
- Sergeant shares a “bad” sensor with another Sergeant (i.e., cross-enclave)
  - “Bad” = one that causes information leakage or that fails to report problems it should have found.
- Sergeant accepts Sensor recommendations from other Sergeants but never contributes any Sensor recommendations.

4.4 Supervisor Trust Evidence

- Revoked or false credentials

5.0 Reputation Calculation

Researchers often recommend that trust be calculated for each type of interaction [8]. For example, an e-commerce vendor may not be good at shipping products promptly, but may provide high-quality products and a generous return policy. For each of these, there is an interaction (product returns) for which we want to measure the quality of service (generous). Their quality of service in one of these areas is not in any way reflective of their quality of service in the other areas, and the type of quality metric that we are interested in may be different as well (generous vs. high-quality vs. fast)

In CID, the interaction that we care about is not the low-level interactions shown in section 4, but is rather the higher-level interaction to "defend the infrastructure", and the quality of service we are interested in measuring is "high integrity". Any indication of lack of integrity is of significant interest in a security application, regardless of the context. Because of this, all trust indicators for a given entity should be incorporated into a single trust value. Similar lower-level interactions can be combined to calculate multiple components of the trust value that can be weighted to reflect the importance of that component to the overall trust (i.e., integrity) value.

Because trust changes over time, only evidence gathered in the last delta time period or in the last *n* interactions should be included in the calculation. The delta time period or *n* interactions should be configurable. This limitation will prevent former good behavior from camouflaging current bad behavior. The limitation should be applied to the components from which the overall trust value is calculated.

6.0 Reputation Decisions in CID

This section considers, for each type of CID entity, how CID should respond to a low trust rating.

6.1 Response to a low Sensor trust rating

- Rather than tracking the reputation of Sensors, Sensors will simply be taken out of service (reported and terminated). There are many Sensors and new ones can be easily created.

6.2 Response to a low Sentinel trust rating

- Report the Sentinel to the Sergeant.
- The Sergeant should change the geography (cf., a loose itinerary) until the problem is resolved.

6.3 Response to a low Sergeant trust rating

- Low trust rating with human Supervisor: If we assume that the sergeant's logic is prescriptive rather than adaptive, then problems can be proactively avoided by testing the code and policy statements prior to deployment. If this has been done, then the sergeant isn't going to go "bad" on its own without having been hacked. To avoid this, the

Sergeant should be installed on a trusted platform. CID will be protecting the sergeant's host as well

- In the cross-enclave case, other Sergeants will refuse to interact with a Sergeant that has a low trust value, so the Sergeant will lose the cross-enclave sharing that would have benefitted its enclave and Supervisor. There should be a feedback loop to the offending Sergeant's Supervisor, perhaps via notification by the Sergeant that calculated/discovered the low trust rating and therefore refused to interact.

Bibliography

[1] W.M. Maiden, J.N. Haack, G.A. Fink, A.D. McKinnon, E.W. Fulp, "Trust Management in Swarm-Based Autonomic Computing Systems", *Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*, pp.46-53 , 2009.

[2] T. Grandison and M. Sloman, "A Survey of Trust in Internet Applications", *IEEE Communications Surveys and Tutorials*, vol.4, no. 4, pp. 2-16, 2000.

[3] Uwe G. Wilhelm, Sebastian Staamann, and Levente Buttyán, "On the Problem of Trust in Mobile Agent Systems", *Proceedings of the Symposium on Network and Distributed System Security*, pp. 114-124, 1998.

[4] M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized Trust Management", *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 164-173, 1996.

[5] T. Grandison, *Trust Management for Internet Applications*, Ph.D. Dissertation, University of London, England, 2003. Available at http://pubs.doc.ic.ac.uk/trust-managem-for-internet-app/trust-managem-for-internet-app.pdf

[6] K.-J. Lin, H. Lu, T. Yu, and C. Tai, "A Reputation and Trust Management Broker Framework for Web Applications", *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Service*, pp. 262-269, 2005.

[7] K. Seamons, T. Chan, E. Child, M. Halcrow, A. Hess, J. Holt, J. Jacobson, R. Jarvis, A. Patty, B. Smith, T. Sundelin, and L. Yu, "TrustBuilder: Negotiating Trust in Dynamic Coalitions", *Proceedings of the DARPA Information Survivability Conference and Exposition* (DISCEX'03), vol. 2, pp. 49-51, 2003.

[8] I. Dionysiou, *Dynamic and Composable Trust for Indirect Interactions*, Ph.D. Dissertation, Washington State University, School of Electrical Engineering and Computer Science, Pullman, Washington, 2006. Available at http://research.wsulibs.wsu.edu:8080/dspace/bitstream/2376/551/1/i_dionysiou_072406.pdf.

[9] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", *IEEE Trans. Knowl. Data Eng*., vol. 16, pp. 843-857, 2004.

[10] Lei Huang, Lei Li, and Qiang Tan, "Behavior-Based Trust in Wireless Sensor Networks", *APWeb Workshops*, pp. 214-223, Springer-Verlag, 2006.

[11] Jereme N. Haack, Glenn A. Fink, Wendy M. Maiden, David McKinnon, and Errin W. Fulp, "Mixed-Initiative Cyber Security: Putting humans in the right loop", Presented at *Mixed-Initiative Multiagent Systems Workshop*, 2009. Available at http://u.cs.biu.ac.il/~sarned/MIMS_2009/papers/mims2009_Haack.pdf.

[12] Trusted Computing Group, http://en.wikipedia.org/wiki/Trusted_Computing_Group.