Combining Trust and Behavioral Analysis to Detect Security Threats in Open Environments

Owen McCusker
Sonalysts, Inc.
215 Parkway N.
Waterford, CT 06333
USA

mccusker@sonalysts.com

Joel Glanfield
Dalhousie U., Computer Sci.
6050 University Ave.
Halifax, NS, B3H 1W5
CANADA

glanfield@cs.dal.ca

Scott Brunza
Sonalysts, Inc.
215 Parkway N.
Waterford, CT 06333
USA

scottso@sonalysts.com

Dr. Carrie Gates
CA Labs
One CA Plaza
Islandia, NY 11749
USA

carrie.gates@ca.com

Dr. John McHugh
Dept. of Comp. Sci
U. of North Carolina
Chapel Hill, NC 27599
USA

mchugh@cs.unc.edu

Diana Paterson
Dalhousie U., Computer Sci.
6050 University Ave.
Halifax, NS, B3H 1W5
CANADA

paterson@cs.dal.ca

ABSTRACT

Open computing environments are under a deluge of network attacks from complex threats. These threats are distributed, decentralized , dynamic, and operate over multiple timescales. Trusted Computing environments provide a means to manage cryptographic identity and authentication operations in the form of static assertions, but were not developed to provide complete end-to-end security for heterogeneous environments such as the NATO Architecture Framework (NAF). There is a gap in the contextual understanding of trust that reaches beyond identity to the behavior of that identity. The challenge in deriving trust, and ultimately risk, from network behavior is that it is inherently subjective compared to identity.

Trust is defined in the Webster dictionary as the "assured reliance on the character, ability, strength, or truth of someone or something" [1]*. When we trust a person there is the notion of identity; e.g., family, and the implied context of trust. Structural identity alone cannot be used to define the overall measure of an entity's trust; the notion of behavior must be taken into account. Trust then becomes a layered concept that can be realized by a number of perspectives including an object's identity along with the behavior of that object. In assessing the trustworthiness of an entity; e.g., host, within a complex enterprise, a cyber defense strategy should take into account various signals regarding identity and behavior that promote an attestation of a digital "self and non-self".*

In this paper we define behavioral-based trust of hosts derived from aggregated network behaviors, which offers a model to bridge this gap and provide a layer of trust that can be used in open environments. We describe a model and approach through which a detection capability can derive trust, and rate the trustworthiness of hosts, based on aggregated network behaviors. This approach is rooted in the context of a global/enterprise

[1] http://www.merriam-webster.com/dictionary/trust

identity management and cryptographic key management (IdM/CKM) which serves as a bridge between the sensor network and the user/administrator/ISP. It offers a mechanism through which an understanding of trust can be derived from aggregated behavioral analysis of network flow data. Our unique view into network behaviors can be used to provide a basis for language to define the various behaviors that threats exhibit over time. We conclude that a more formal model of trust is needed that couples identity with behavior along with the identity of the user of a computer.

1.0 INTRODUCTION

The North Atlantic Treaty Organization (NATO) is faced with the increasing need to support international operations that leverage the use of Enterprise Architectures. NATO Network Enabled Capability (NNEC) is an integral program focused on meeting these needs [1]. The ubiquity of these net-centric information systems is realized by the connectivity of hand-held technologies, operated and managed by users in the field, to backend mission support systems managed by tens of thousands of administrators. Each officers' transactions cross into the operational jurisdiction of multiple ISPs that deploy a number of commercial off-the-shelf hardware systems, which in turn provide symbiotic Internet services; *e.g.*, DNS and IPv6. Today each hardware system and service within the Service Oriented Architecture (SOA) provide an independent notional thread of trust within the open fabric of the Internet.

The European Union (EU) has outlined the need for international cooperation in section 2.6 of draft Deliverable 3.1B focusing on the need to develop international security standards [2]. The United States (US) in Point 9 of the Cyberspace Policy Review noted that "In collaboration with other [Executive Office of the President] EOP entities, develop a framework for research and development strategies that focus on game-changing technologies that have the potential to enhance the security, reliability, resilience, and trustworthiness of digital infrastructure." [2]

In order to realize the defense of these complex systems a multi-layered approach is required that consists of: a) hardened software and operating systems, b) trusted hardware, c) trusted tokens, d) distributed decentralized behavioral sensors within application software, hardware and networks, e) a distribute decentralized IdM framework that associates electronic identifiers (and possibly organizational identities) with hardware tokens, and f) a distributed decentralized IdM framework that associates human users with unique tokens.

Trusted hardened software and hardware is required to reduce the number of exploitable weaknesses, sensor networks to detect the presence of malicious software or malicious user behavior, and various identity management and authentication frameworks to facilitate the application of access control technologies to software, data, and users. One such gap is the formalized notion of trust found within a Digital Provenance (DP) [3].

Ultimately, the definition of DP must reach beyond notions of trusted hardware and software and extend into the realm of both digital and human identities; *e.g.*, company registries, legal persons and physical person. DP then becomes an aspect of Information Assurance and is used as a method in managing risk in Enterprises. Information assurance (IA) is the practice of managing risks related to the use, processing, storage, and transmission of information or data and the systems and processes used for those purposes. In this paper we will scope DP to be realized by hardware and software through which trust can be initially derived.

Trusted Computing exists to ensure trust within a given entity, and does not address end-to-end security [4]. Trusted computing, encompassing hardened software and hardware, is intended to ensure a system performs a given set of well-defined functions correctly, even when an attacker attempts to coerce it to do otherwise. However trusted computing does not imply the computer is "trust worthy" from the perspective of the user, i.e.,

[2] http://media.pqs.io/pub/papers/ThinkTrust/20100126-TT-D3_1b-Consult-Synaptic-Input-Part2-RapidEvolve.pdf

trusted software does not imply it can be trusted by all users to behave in their best interest. For example, a Government may require “trusted software” to report on civilian activities back to the Government, even when the user attempts to prevent it. From the perspective of the user, this software would be accurately considered spy-ware. This can be expressed in several forms, such as “software license management”, where the computer enforces the usage policies of a vendor, and not the computer’s owner.

Access control systems, implemented on trusted computing platforms, require the availability of identification and authentication services. These may be provided by assertions/declarations/guarantees made by identity management service providers, hardware tokens, biometric readers, and/or challenge/response schemes. Cryptographic mechanisms, such as message authentication and digital signatures, running on hardened platforms allow a chain of assertions to be electronically validated. Access control policies may require a given set of credentials/assertions to be provided and be cryptographically validated before software can perform a privileged operation.

Modern X.500 based PKI IdM/CKM systems were designed as predominantly offline systems for making identity assertions. These offline architectures do not suggest an obvious method for bridging online behavioral trust with their offline static identity assertions. Summarizing the content of various statements made (2009) by senior U.S. National Institute of Standards and Technologies (NIST) staff at the NIST CKM Workshop [5], there exists a gap between today’s CKM/public key infrastructure and the requirements needed to vastly improve ICT security. Such designs should be highly available, fault tolerant, and support accountability, auditing policy management which we observe are requirements leaning towards behavioral trust.

Behavioral analysis extends a static assertion checking model with history. This history is used to determine if the software, hardware or users performing an action has a known dirty track record and is considered a high security risk. This is a useful risk management strategy that limits exposure but cannot itself guarantee that the software/hardware/user will not act maliciously in the next instance. What large systems need are formal strategies to derive trust from a new online global identity management and cryptographic key management (IdM/CKM) architecture, such as the one proposed by Synaptic Laboratories [6] which satisfies many of NIST’s 2009 CKM requirements, and by creating and synergistically integrating a dynamic notion of trust derived from Network Behavioral Analysis (NBA). Behavioral-based trust has been researched as a enabling technology to provide trust in an open environment such as NNEC [4, 7].

The complexity of cyber threats has steadily increased in recent years evolving into distributed threats operating over large time scales. Our detection models and defense strategies are still tuned for single ingress points. In 1987, Denning proposed an intrusion detection model that focused on the identification of network attacks directed toward a single host [8]. The threat at that time comprised attackers attempting to gain remote access to a host. Todays threats are often distributed, decentralized, dynamic, and operate over multiple timescales. In addition to a renewed rigor in applying traditional information assurance methods, we require new behavioral models that are capable of detecting known and new threats to our network infrastructure, facilitating both deterrence and correction [9]. Considering the increasing size and complexity of NATO networks and the threat space, we note that there is no single algorithm or strategy that can detect the wide range of known and emergent threats that exist. In response to this complexity, we join the migration from developing solutions that are reactionary to developing those that are more proactive, by leveraging current and historical information. Another threat to NATO-like environments is the rate-of-change of contemporary cyber threats, as it negates the effectiveness of misuse-based detection strategies that use known attack signatures as a method of detection. NBA, as used in the agent-based system CAMNEP, is an approach towards addressing this challenge. But even systems like CAMNEP can be hindered in their ability to detect real-time threats due to the volumes of both current and historic traffic, an issue that is of concern to those monitoring NATO networks [10, 11].

In this paper we address the derivation of trust about a host measured by a sensor performing aggregated behavioral analysis in the context of a Digital Provenance. Behavioral monitoring can happen at all levels, from application to network, and this paper will focus on the network. The goal is to offer the measurement of behavioral trust further enabling and strengthening end-to-end security needed to foster an understanding of risk to operations in complex environments such as NAF. A key aspect of our approach is the ability to find malware behavior without infringing on the privacy of individuals. We review a number of enabling concepts, technologies and models such as Digital Provenance, Trust Computing, Network Behavioral Analysis, Behavioral Trust, and digital "self, non-self". We then propose a model for the derivation of subjective behavioral trust, using aggregating behavioral analysis and suggests that behavioral security might be integrated into IdM/CKM both internally, and as a clearing house. Lastly, we highlight future research leading to development of a behavioral Ontology. Our approach is of benefit to NATO analysts since it addresses the issues noted above; *e.g.*, volume of data, complexity of networks, and the rate at which cyber-threats change. It is based on the insight that a common metric is needed to provide an understanding of risk to the analyst.

2.0 RELATED WORK

In this section, we review related work on digital provenance, behavioral analysis and trust. We present important works on trust and behavioral analysis, intrusion detection leveraging network flow and cyber situation awareness. This section contrasts the evolving threat with the models that were used in establishing existing detection technologies.

2.1 Trust Models

A large number of trust models have been created, most of which define trust statically, *i.e.*, in terms of the identity and authentication of single entities within a network information system. These "fixed evaluation schemes contradict the subjective nature of trust" and do not reach out to the operational needs and vulnerabilities inherent in open systems [7]. Even though a system can be identified and authenticated by cryptographic means, it still can be compromised by threats like a botnet. In the context of trust, its behaviors can be used to identify, and behavioral-based trust can be used to manage the overall trust of the entity. Open and heterogeneous computing environments need a more dynamic formalization of trust, combined with trust derived from identity.

Weth and Bohm have provided a unifying notion of trust based on behavior [7]. In this approach, they create a formal representation of behavior-specific knowledge about an entity, *e.g.*, a user in a virtual environment. They formulate the notion of both first and second-hand knowledge based on the experiences of other entities. Their model defines four types of behavior-specific knowledge.

1. **Feedback:** a feedback of an entity's rating, μ_{ratee}, of interaction performed by another entity, μ_{rater},

$$feedback = (\mu_{rater}, \mu_{ratee}, \phi, \psi_{\phi}, \tau, \sigma, \epsilon, \nu) \tag{1}$$

In the case the rater entity, μ_{rater}, scores the ratee, μ_{ratee}, based on the context, ϕ, of the measurement of the value measured, ν. The value of trust is context-specific and dependent on the facts, or knowledge, supporting the measurement. They define the ϕ, context of the interaction, is the set of all facts or circumstances that surround an interaction. The effort, ϵ, is a measure of the estimated cost of behavior by an entity, both good and bad. The time dependency, τ, take into account the trustworthiness of an entity at t_1 versus a measurement of trustworthiness at t_2. Lastly, certainty, σ, is a measure of accuracy of trust associated by a entity, ν_{rater}, and is measured in the interval $[0, 1]$.

2. **Recommendation:** is the opinion from one entity about a previous behavior of another entity, and is typically an aggregation of experiences,

$$recommendation = (\mu_{recommender}, \mu_{recommendee}, \phi, \psi_{\phi}, \tau, \sigma, \nu) \tag{2}$$

3. **Reputation:** is the general opinion of a whole population of entities, μ, regarding specific behaviors of a single entity, and is typically an aggregation of experiences.

$$reputation = (\mu, \phi, \psi_{\phi}, \tau, \sigma, \nu) \tag{3}$$

4. **Trust:** is the belief held by an entity that another entity will behave as expected in future interactions, and is based on recommendations and direct experiences regarding an entity, or that entities reputation.

$$trust = (\mu_{truster}, \mu_{trustee}, \phi, \psi_{\phi}, \tau, \sigma, \nu) \tag{4}$$

Haldar and Franz have well noted that although “Trusted Computing is a solution to the trust problem regarding identity, it does not address the problem of end-to-end security” [4]. They have created a different behavioral model called “remote semantic attestation,” which derives improved assurances of trust from the execution of code in a Trusted Virtual Machine e.g. java virtual machine, and is based on a number of behavioral properties that address the end-to-end security needs. In [4], it is noted that there are two fundamental views on trust; first, trust in terms of an entity’s integrity and authentication, and second, how an entity is behaving within a given environment. Behavioral-based trust models are a common way to determine trust of an entity in an open environment [7].

In 2009 at the National Cyber Leap Year, a broad group of cyber-security, immunological biologists, and other researchers met to share ideas both past and present, putting together a notion of trust. In their report, end-to-end trust is defined as a “set of technologies, behaviors, implementations, and infrastructure that, when used consistently, can enable a predictable level of trust” [3]. In order to assess trust in these complex systems a formal understanding is needed.

In 1994, Marsh discussed a computational formalism regarding trust that provided a basis through which trust can be measured [12]. In May 2009, Rehak leveraged Marsh’s work to aggregate and share trust within an agent-based IDS called CAMNEP (as previously discussed). A layered defense approach offers the most flexible strategy for end-to-end trust. In all cases, including [7], computational trust has been viewed as a subjective metric driven by the operational security policies held within specific network environments.

In this paper we consider trust as a layered concept that has its core in Trusted Computing. In our model we define trust in terms of entities representing hosts, instead of users in a virtual social environment as formalized in [7]. We build off of the model created by Weth and Bohm applying it to network security, and simplifying some of their definitions. In our model, trust of a entity; *e.g.*, host, $trustee$, is a measure managed by a truster; *e.g.*, Enterprise (NAF). Context with the measurement of feedback is driven by network events and data as input representing facts and/or knowledge about the entity; *e.g.*, hosts. Recommendations come in the form of trust derived from aggregated network behaviors measured in one or more locations by one or more sensors, Feedback is closely associated with the network events and data received by sensors. We did not yet take into account $effort$, in our measure of trust in feedback, nor the concept of $facet$ as defined in [7]. We chose not to incorporate the use of Internet reputation which can be made available from recommendations from systems like McAfee TrustedSource$^{\text{TM}}$[13]. Our contribution to behavioral trust is a method used to derive it from aggregated network behaviors.

2.2 Digital Provenance

Digital Provenance has been identified as one of the key game-changing ideas that could affect the cyber security domain drastically over the next decade [3]. Cheney *et al.* give a hindsight perspective of provenance from a futuristic view [14]. Although hypothetical in a nature, the suggestion is that provenance, if taken seriously, can become ubiquitous. But such ubiquity will not come without a cost. Technology must adapt to allow provenance to be recorded such that it will be considered as essential a resource as physical network components themselves. Besides the technological implications of implementing provenance, there are many other subjective, objective, and semantic issues that must be considered.

Yet the implementation of provenance provides a framework within which one could formulate a trust model, creating trust-based security mechanisms [12]. This can help clarify the notion of human trust in spite of its subjective nature [7]. Gray et al. (2003) provided an example of such a mechanism by applying small world concepts to mobile ad hoc networks [15]. The small world concept "describes the tendency for each entity in a large system to be separated from any other entity in the system by only a few steps". One of the main problems they address is how an entity can assign a trust value to another entity never before encountered. By applying small world concepts, an entity is able to prompt already-encountered entities for their experience with the current unknown entity, and thus derive a trust value based on the experience of others. We refer the interested reader to the paper for more details. The main point here is that frameworks can be constructed within which trust can be computed by a detection system and assigned to a host.

Rehak *et al.* provide a system that makes trust decisions from interactions between detection, aggregation, and incident agents [10]. Trust in CAMNEP uses a "network flow's identity and context to define a feature space thats specific to each anomaly detection agent". It is the detection agents that detect anomalies and compute trust values based on their respective models, but it is important to note that the job of the aggregation agents is to aggregate those trust values. Thus, a collective opinion is formed regarding the trustworthiness of network flows. Our concern regarding trust aggregation is that the subjective nature of trust suggests that it should be policy driven since network information systems are heterogeneous. There will be times when different policies are driven by different member countries, but the behaviors remain the same. Haldar and Franz support this view by stating that an entity's authentication should "... include verifying or proving that its behavior conforms to a required security policy" [4]. It is difficult to divorce trust from behavior. Hence, our aim is to leverage the notion of Digital Provenance in deriving behavioral trust, because although we realize the need to consider hardware and policies, we must also consider the past behavior of network entities when computing trust-based attribution. Digital Provenance provides a framework within which a common semantic definition of digital artifacts can be given, where such artifacts are used to measure risk to operations in near real time. Although Haldar and Franzs work is limited to the behavioral aspects of programs, as we consider network entities such as the end host, their work brings to mind the need to formalize past behaviors and interactions in an effort to overcome fixed evaluations schemes (see also [7]).

Martin and Lyle argue for a synergistic relationship between digital provenance and trusted computing [16]. They feel there is overlap between the two fields regarding "both the goals and technologies in achieving them". They present a notion of a trusted provenance and present an attestation-based provenance architecture. This architecture provides a set of distributed services that process results and attest to a remote provenance store. We would propose that behavioral trust derived from network behaviors can be layered within the trusted provenance providing a complementary addition to a trusted provenance.

2.3 Behavioral Analysis

There exist two broad categories of network defense technologies: misuse detection [17, 18] and anomaly detection systems (AD) [19]. In the past decade, misuse-based detection systems have been a mainstay for organizations to secure their network infrastructure. The effectiveness of misuse-based technologies is impacted greatly by the fast rate at which cyber threats modify both their structural and behavioral characteristics. These systems cannot keep up with adaptations in malware. AD offers the ability to accelerate creation and propagation of signatures in misuse detection systems.

AD systems are driven by the use of various heuristics in developing an understanding of normal network behavior and using this model to detect anomalies. It is this ability to baseline normal behavioral patterns that allows such systems to adapt to emergent threat behaviors. NBA provides capabilities that focus on network behaviors and the structural properties of threats.

Traditionally, both of these perimeter-based technologies derive events from a single ingress point of an enterprise's network. In order to scale to the needs of complex threats realized on open systems such as NAF, distributed defense capabilities are needed to bring together alerts that define threatening behaviors derived from multiple ingress points. Different enabling technologies are needed to manage the volumes of data originating from large enterprises and to break through the privacy barriers of multiple organizations, preserving privacy, all while fusing network events and data.

One such enabling technology is aggregated behavioral analysis. Some work on the detection of botnets discusses this approach [20]. Data is fused from multiple sensors, after which a feature characteristic is mapped to each network object. The feature characteristics are then projected into multiple hyperplanes where an analyst can decide which hyperplane division is best suited to the problem at hand – in this case, botnet behavior detection. Once the appropriate division is constructed, a correlation function is responsible for assigning a score to each network object. Within a controlled environment, the authors demonstrate that a division and correlation can be found that identifies bot-like behaviors. Thus, an analyst can consider various behaviors of a network object, but also consider aggregating such behaviors to describe and detect more complex behaviors that may vary over time and space. Normal aggregated behaviors are visualized and highlighted in Figure 2. These behaviors are aggregated over a monthly timeframe and contained in a behavioral feature space.

Consistent behavioral patterns have also been discovered upon the inspection of live data sets over long time periods and across different data sets (Figure 2). Aggregated behavioral analysis provides an enabling technology through which weak-signal threat detection and over-the-horizon cyber-defense solutions can be created.

Soldo *et al.* have used behavioral analysis for attack forcasting and predictive blacklisting using aggregated network data sets [21]. In that work, a Dshield dataset is analyzed for temporal changes in behaviors that can be used to identify patterns then used for blacklisting.

As mentioned in Section 2.1, Rehak *et al.* introduced a behavioral analysis system called CAMNEP [10]. The system aggregates trust scores produced by several anomaly detectors. Even though this type of system can be hindered in its ability for real-time detection due to leveraging the volumes of both current and historic traffic during analysis, it takes a step in the right direction towards combining computational trust with behavioral analysis.

These approaches have implications when considering the amount of network data that needs to be stored since we are considering aggregated behaviors as opposed to classifying raw flows. It also has implications in the privacy domain since we discuss behaviors as opposed to discussing identities. One of the challenges of the current state of anomaly detection is the inability to "... mitigate slow, stealthy and sophisticated attacks" [11].

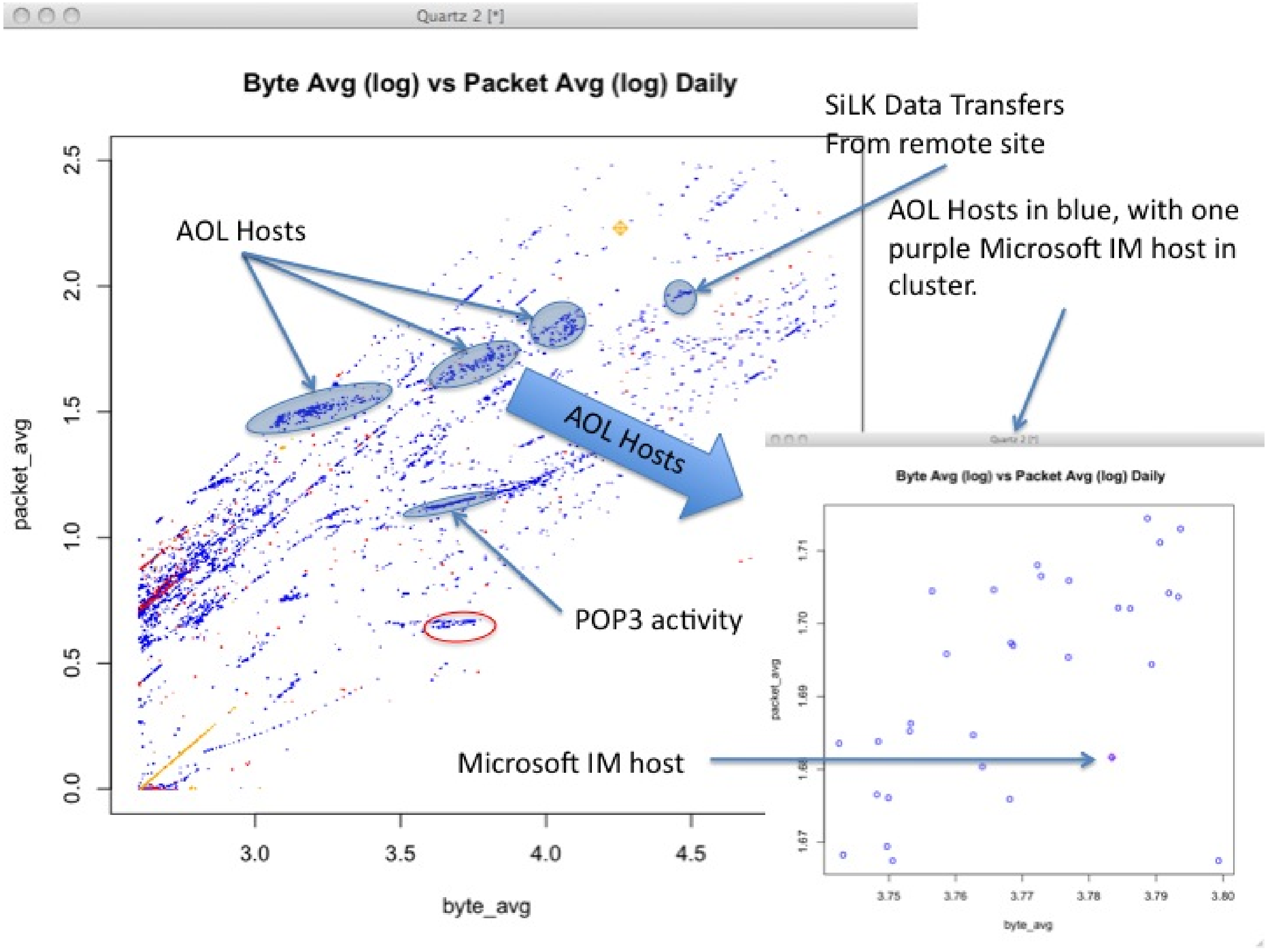

Figure 1: Clusters of normal network patterns

The work done in [20] is an example of how behavioral analysis can detect such threats, and then drill into in the raw network data to further analyze the threat patterns. More work needs to be done in the sharing of such behaviors to determine whether mitigation can be achieved in a reasonable time.

2.4 Digital Self-Nonself for the Enterprise

Forrest *et al.* have inspired self-nonself concepts by studying Biological Immune Systems (BIS) and by first applying them to the detection of abnormal Unix processes on a host [22, 23]. Dasgupta has studied immune signaling mechanisms and modeled their application to cyber defense technology [24]. These mechanisms have been used in a two-signal view of self-nonself in immunological systems [25]. In the context of immune signaling mechanisms, BIS can be realized as an adaptive mutli-layered defense system.

Dasgupta states that "...in the immune system, signal diffusion and dialogue are noticeable as two kinds of communication schemes" [24]. He defines signal diffusion as the message (*e.g.*, a "recommendation" [7]) that is passed between immune system components, where dialog represents a continuous exchange of signals between these components. What is important with respect to trust and trustworthiness is the consideration

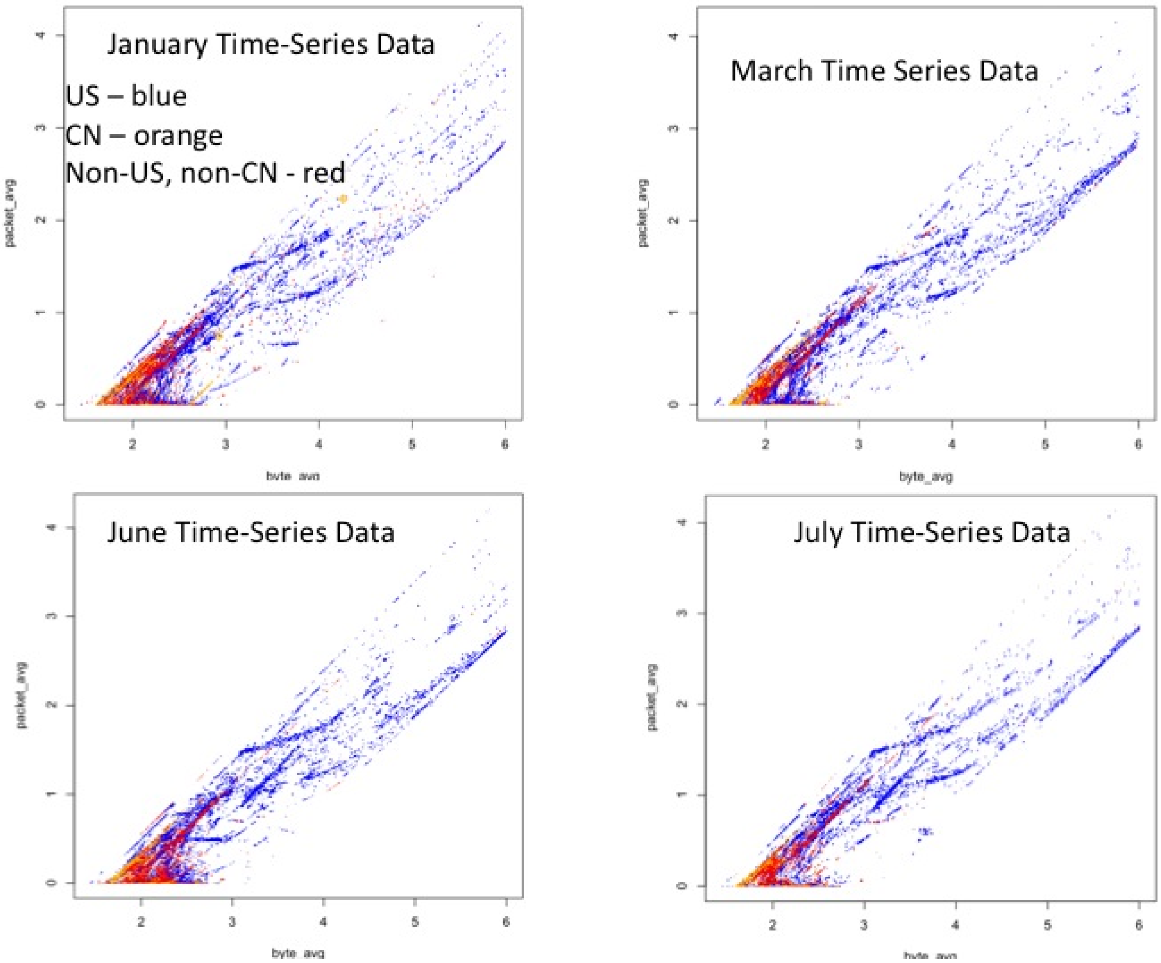

Figure 2: Consistent behaviors exhibited over monthly time intervals

of context when determining immune sensitivity, since the deriving of trustworthiness should also consider context (*e.g.*, network policies) [7].

Instead of comparing the human body's BIS as an analog to a single host, we can view the human body as a set of independent, although related, entities (*e.g.*, cells coming together to form self). The digital analog of BIS can then be the application of self-nonself to a complex network of devices in a pervasive environment, where signaling is not only found within the device (*e.g.*, host, smartphone) but also across devices within a network. In this case, self is understood not only by the identification and authorization of self via cryptographic means, but also by the systematic classification of both structural and behavioral patterns perceived over the network.

Trust, from the perspective of the Enterprise, can then be seen as a metric used to determine self and nonself, where each are derived from various types of signaling within a multi-layered, distributed system, and where signaling, in the context of trust models, is a form of two shared behavioral types: "feedback" and "recommendations" [7]. In our system, we have investigated behavioral self in terms of aggregated host behaviors that are captured over multiple time periods, and then observing a consistency within an enterprise (see Figure 2). In the Conclusion section, we identity the future need to focus more on the systematic classification and identification of behavioral primitives in the form of an Ontology, which would then foster the creation of narratives [26] that describe threats.

2.5 IdM/CKM Architectures Capable of Supporting Digital Provenance

In the introduction we stated that behavioural analysis extends a static assertion checking model with history. In distributed, decentralised, net-centric information systems, it is not possible (or necessarily desirable) for every device/process to maintain history on the behaviour of identities that it has interacted with. It seems natural to explore if the historical behaviour of an identity could be synergistically integrated with the infrastructure providing assertions regarding identity.

As previously mentioned, modern X.500 based PKI IdM/CKM systems were originally designed as predominantly offline systems for the purpose of making static identity assertions and enabling authenticated/secured communications. These offline architectures do not inherently suggest an obvious scalable method for achieving online behavioural trust assertions/queries. Furthermore, when we take a closer look at the specifications of the X.509 architecture, we find that the civilian PKIX architecture cannot guarantee the unique assignment of a public identifier such as a website address or email account to one certificate/individual/party. Today's civilian PKI has 20+ autonomous Root Certificate Authorities, each of which are a system-wide single point of trust failure for identities on the Internet [27–29]. According to a recent 2007 survey, approximately 86% of fraud happens by management level staff against their own organization [30]. Part of the identified problem is that senior management are often able to circumvent the internal security mechanisms intended to prevent fraud. It is simply not clear what level of confidence can be assign to assertions made by civilian PKIX systems. Today, no standard mechanism exists by which multiple organisations can attest to an identity to increase our confidence on the static assertions that we are validating.

In 2009, NIST stated that "numerous problems have been identified in current key management methodologies". Summarising the content of various statements made by senior U.S. NIST staff at the NIST CKM Workshop [5], there exists a gap between today's CKM/public key infrastructure and the requirements needed to vastly improve ICT security. According to NIST staff, such designs should be highly available, fault tolerant, secure against destructive attacks, scalable to billions of users/devices, be secure against quantum computer attacks and not use public key technologies. Additionally they must support accountability, auditing policy management [5] which we observe are requirements leaning towards behavioral trust.

Online IdM/CKM systems require the service provider(s) to maintain state so they can store pair-wise unique symmetric key material and make positive real-time assertions concerning the validity of an identifier. The presence of state in online IdM/CKM architectures that can be queried by a remote user suggests that these designs can be extended to support behavioural trust capabilities, both internally and as a clearing house for external assertions regarding credibility.

We point to the m-1 secure symmetric key distribution protocol proposed in 1976 that exploits m key distribution centers [31] and Synaptics IdM/CKM proposal which extends that result [6]. We agree with the Synaptic authors and assert that new IdM/CKM architectures should distribute the execution of each provisioned service across m autonomously owned/managed service providers to mitigate insider fraud/attacks, where $2 < m < 7$. As the Synaptic authors also point out, these architectures permit the principles of separation of powers and checks and balances to be embodied for services provisioned to the community [32].

3.0 A DETECTION MODEL INCORPORATING TRUST

This paper focuses on the derivation of behavioral trust based on aggregated network behaviors calculated from network flow. This trust model is based on the detection model described in [20]. The components of this model are captured in Table 1. This model is driven by a behavioral-based fusion architecture presented in Figure 3. The architecture uses behavioral analysis functions that process and fuse network flow data creating a n-dimensional behavioral feature space.

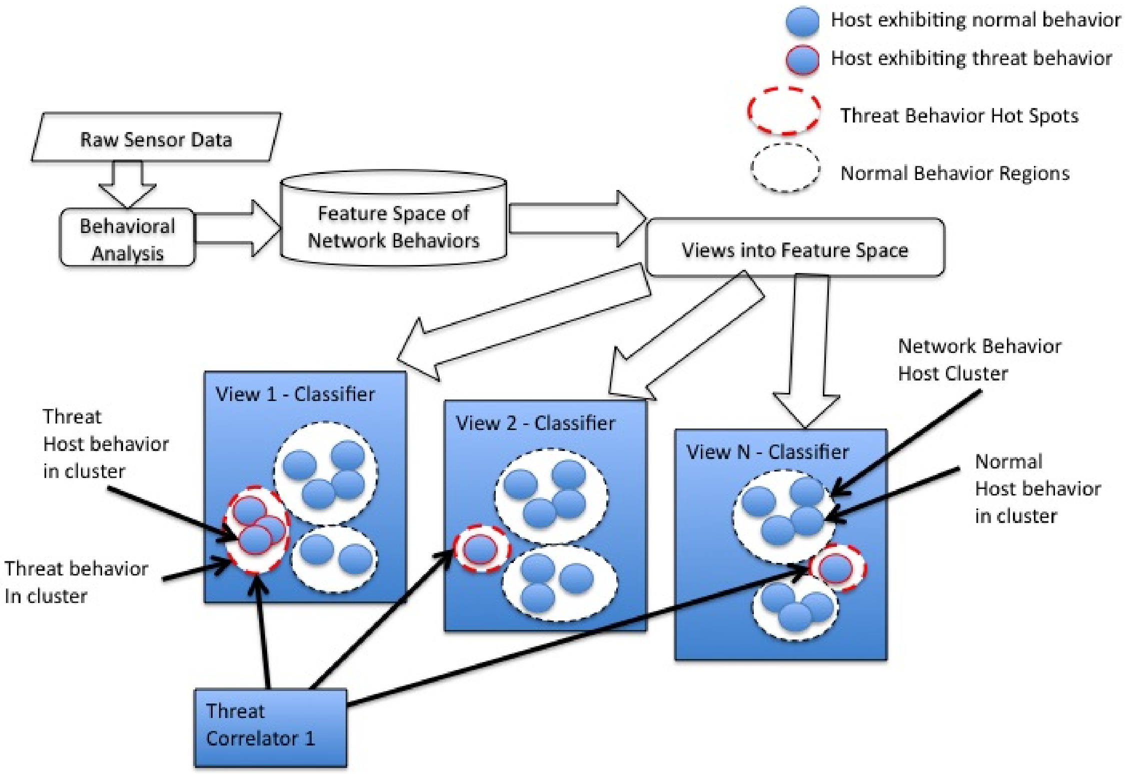
Host exhibiting normal behavior
Host exhibiting threat behavior
Threat Behavior Hot Spots
Normal Behavior Regions
Raw Sensor Data
Behavioral Analysis
Feature Space of Network Behaviors
Views into Feature Space
View 1 - Classifier
View 2 - Classifier
View N - Classifier
Network Behavior Host Cluster
Normal Host behavior in cluster
Threat Host behavior in cluster
Threat behavior In cluster
Threat Correlator 1

Figure 3: Overview of a Behavioral Analysis Architecture

The detection model processes network data and events, d_O, using network sensors, S, grouping the data by network objects, O, *e.g.*, hosts. Instead of being an alert-centric model, this model focuses on identifying threats, and threat behaviors in terms of higher-grained network objects; *e.g.*, hosts. Note, that in the case of network flow, the network data represents a tuple where $tuple_d = \langle d_1, d_2, \ldots, d_n \rangle$. Network data from sensors is inherently non-numeric and must be transformed into numerical form represented by td_O and their tuples $tuple_{td} = \langle td_1, td_2, \ldots, td_n \rangle$.

Various behavioral analysis functions, $bf_{analysis}$, operate over the transformed data, $tuple_{td}$, creating an n-tuple of behavioral features values, ν_{bf}, associated with each object processed, O. The behavioral feature value, ν_{bf}, represents $feedback$ in the overall model, where,

$$bf_{analysis}(tuple_{td}) = \langle \nu_{bf1}, \nu_{bf2}, \ldots, \nu_{bfn} \rangle = tuple_{\nu bf} \qquad (5)$$

These behavioral value tuples, $tuple_{\nu bf}$, are then input into a shared sample space or behavioral feature space, $FS_{\nu bf}$. Behavioral values can represent the results of simple heuristics such as statistically determining if network flows associated with a host are incoming outgoing. In the future, they also can represent a metric resulting from a distance measure. A distance measure may be a good measure when tight clusters of behaviors are exhibited like the AOL cluster in Figure 1. A distance measure of the host, O_{host} from a centriod of a region, c_{region}, of the behavioral feature space, $FS_{\nu bf}$, can be calculated where,

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{6}$$

If $O_{host} = (h\nu_{bf1}, h\nu_{bf2}, \ldots, h\nu_{bfn})$ and $c_{region} = (c\nu_{bf1}, c\nu_{bf2}, \ldots, c\nu_{bfn})$, then formula 6 can be generalized by defining the Euclidean distance from a to b as

$$d(O_{host}, c_{region}) = \sqrt{(c_1 - h_1)^2 + (c_2 - h_2)^2 + \cdots + (c_n - h_n)^2}. \tag{7}$$

Classifiers are tuned to view the features space using different sets of features or feature-tuples. Within those views behavioral regions are defined and a distance measure is used to score a host within a region based on the centroid of that region. In this context, classification is determined from a set of behavioral features values, ν_{bf}, calculated using various types of methods including: simple statistical based heuristics and distance measurements from known regions.

In the first instantiation of our framework, our basic network object is a host in the internal network. For each host, we post-process any raw data that is available through the sensors to distill a feature characteristic that includes the following as shown in Table 2.

The proposed detection model is used to analyze various behavioral and structural characteristics of network objects, O; *e.g.*, hosts, hostgroups, and subnets. The analysis of these objects involves collecting sensor events and network flow data R from a number of different network sensors; *e.g.*, network flow, NIDS, honeypots, and creating a behavioral sample space, S. The sample space for a specific object O is denoted by S_O. The trust of each behavior, bt, is computed during the analysis. Lastly, the overall trust of a network object is computed (T_O). The framework employs the following major components as described in Table 1.

In our model, an object's trust, T_O, is derived from behavioral trust associated with network behaviors of an object. Each individual behavioral trust value can then be measured from the various network sensors made available in the enterprise. In order to utilize the behavioral feature values, a subjective notion of trust must be collected and applied to the values. These recommendations are in the form of behavioral trust measurements as defined in Table 1. This model is not as complete in the unifying model formalized in [7] where we do not take into account the notion of reputation.

The first way to assign object trust, T_O, based solely on behavioral feature values, would be to take into account the subjective weights associated with each measured network behavior. These weights are determined by the security policy of any given institution. We would then look at calculating a behavior trust value and then summing across all behavioral feature values. This would provide a baseline notional object trust and is formally defined as follows:

$$T_O(1)[0,1] = \sum_{0,n:\nu_{bt}} wt_P(S) \tag{8}$$

where object trust is a value in the range from 0 to 1. Trust is then measured form the summation runs over all behavioral trust values ν_{bt}. Nevertheless, the first degree score fails to factor in the identity of a certain host. This is taken into account in the second degree notion of the object trust:

$$T_O(2)[0,1] = \sum wt_P(S) \cdot identity(O, P) \tag{9}$$

Table 1: Model Components

Element	Description
Sensor (S)	A device providing observables in the form of raw network data and/or network events.
Network Object (O)	An object being tracked, representing either a host, host group, or subnet.
Network Object Data (d_O)	Raw data consisting of sensor events and network flow data.
Transformed Network Object Data (td_O)	Network data and events are usually non-numerical in nature and must be transformed into a numerical format for processing; *e.g.*, average incoming network bytes per flow. This data is used by the behavioral analysis functions ($bf_{analysis}$).
Behavioral Analysis Function ($bf_{analysis}$)	A function that operates over the sample space to create a behavioral feature value (ν_{bf}) for a network object (O).
Behavioral Feature Value(ν_{bf})	A single behavioral feature produced by ($bf_{analysis}$). This represents a single classification primitive.
Behavioral Feature Space($FS_{\nu bf}$)	The feature space used by classifiers and correlaters.
Behavioral Feature N-Tuple (t_{bf})	A set of features (in the form of a record of label-value pairs) describing a network object O. The feature characteristic of a host, for example, may list inter-packet arrival time, outgoing packet size, work-weight ratio, etc.
Behavioral Trust Function (bf_{trust})	A function that operates over the behavioral feature to compute the behavioral trust bt for a network object (O).
Behavioral Trust Value(ν_{bt})	A single behavioral trust value produced by (bf_{trust}). This represents a single behavioral trust recommendation produced by a sensor; *e.g.*, entity. primitive.
Overall Trust T_O	The overall trust of a network object (O) derived from each of its behavioral trust bt values .

where the summation runs over all pairs $identity$ and T_O both contain object O respectively.

The wt_P weight function determines the significance of a particular behavioral feature in the final trust calculation. Note that the weight function takes as input the labeled partition of network objects that was produced by the behavioral trust function as the weight should be placed appropriately depending on the label if it exists.

In this section we summarized our detection model built upon a technique we call aggregated behavioral analysis, that is used to derive a subjective notion of behavioral trust. This trust is driven by the security needs of enterprise which are realized as weighted functions applied to behavioral features values. We later visualize trust in Figure 5 providing a descriptive analysis of our results.

Table 2: Behavioral Features (from [20])

Feature	Description
Port Scatter	Number of unique destination ports
Source Data Transfer	Ratio of total bytes of traffic originating from the host to the total bytes of traffic
Source Data Transfer	Ratio of total bytes of traffic going to the host to the total bytes of traffic
Source Packets	Number of packets origination percentage of the total traffic to incoming
Sink Packets	Number of packets origination percentage of the total traffic to incoming
TCP, UDP, ICMP Bytes	Number of incoming and outgoing bytes (by protocol)
TCP, UDP, ICMP Packets	Number of incoming and outgoing packets (by protocol)
TCP Work Load Index	Ratio of the number of TCP bytes transmitted or received to the number of TCP flows
Social Index	Number of unique hosts a computer talks to
Packet Inter-arrival Time	Total and average packet inter-arrival time

4.0 APPROACH

In this section, we review the approach we taken in the application of behavioral analysis on network flow data captured with the SiLK [3], and the derivation of behavioral trust with qualitative analysis of some of our results.

4.1 Behavioral Aggregation

Aggregated behavioral analysis provides a natural way by which the volume of network data can be reduced, and it provides insight into views of network behaviors over multiple time scales. Behavioral analysis is used to drive the measurement of various layers of trust using a model we have created (described below). We differ from CAMNEP since we aggregate all behaviors, instead of just anomalies, thus separating trust from the use of clustering strategies on anomalies [10]. Another major difference between the systems is that we initially aggregate at the IP Address level. Aggregating at the host level offers the ability to track behavioral changes in a specific host; *e.g.*, a botnet, and to cluster them based on similar behaviors or behavioral changes. Another benefit of IP Address behavioral aggregation is the further reduction of data which allows the detection system to recognize patterns over multiple time scales. Similar to CAMNEP, we leverage the use of trust to further classify the hosts, but, we do not aggregate trust, nor share trust. In the context of our model, trust is a subjective measure and intertwined with the the security policies environment through which it is measured.

In our approach we employ two independent (but related) metrics, i.e., behavioral and trust metrics. Behavioral analysis metrics provide indicators of known behaviors: both normal and anomalistic. Ultimately, the trust of

[3] http://tools.netsa.cert.org/silk

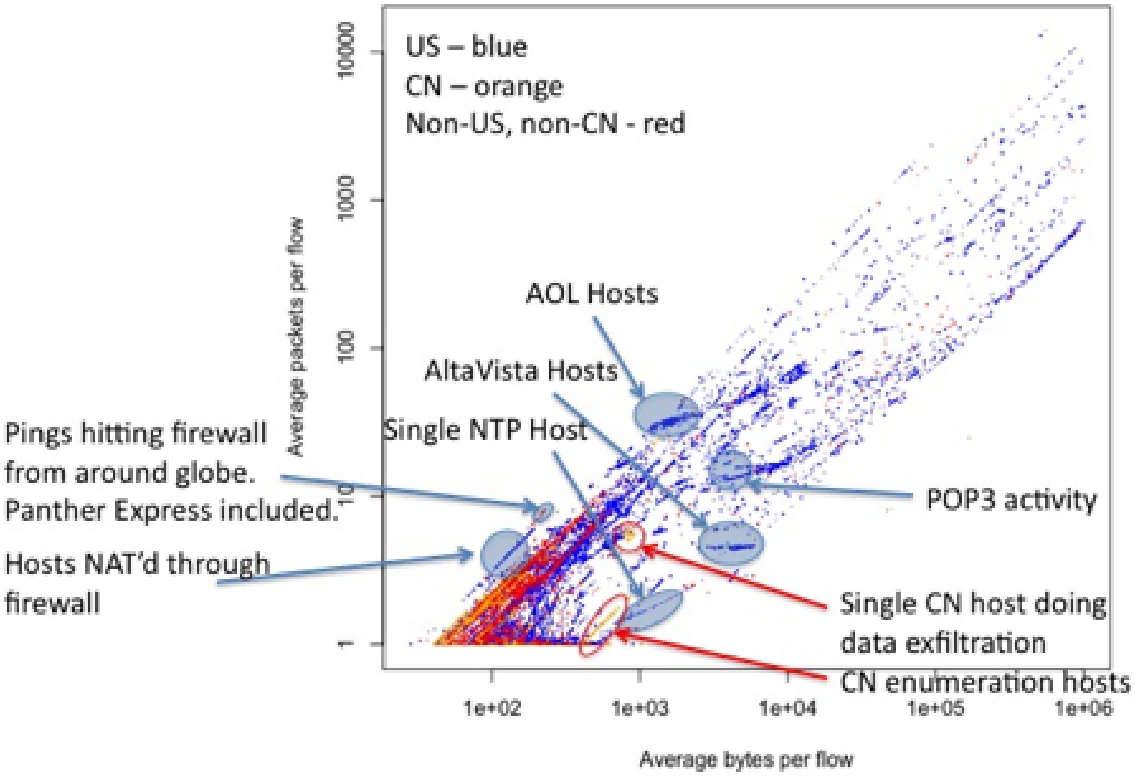

Figure 4: Data exfiltration captured by our system

these behaviors is driven by their context. Examples of trust metrics include source/sink trust; *e.g.*, data exfiltration to specific regions is trusted less, incoming and outgoing byte/packet usage trust; *e.g.*, low packet per flow, small byte payloads scan behavior, incoming and outgoing protocol usage trust; *e.g.*, IP Address using multiple protocols, and regional communication trust.

The overall detection model is made of the following elements: sensors, raw data, and network objects, hosts, features, behavioral functions, aggregated behaviors, classification strategies, classifiers, correlation strategies, correlators, behavioral trust, and trust. Our detection model is based on the Tadda cyber data fusion model [33]. In this model, formal relationships exist between various elements. Sensors provide observables to the fusion engine in the form of network data and events. We have ingestors for network flow, Snort IDS, and Honeytrap, although we are currently focusing on network flow (more specifically SiLK flow data). The model uses the raw data to extract network objects; *e.g.*, hosts, that are used to track aggregated behaviors. Features are extracted into shared memory consisting of an n-dimensional feature space. Behavioral analysis functions are applied to the feature space to measure behaviors and calculate behavioral trust. Classification strategies employ multiple behavioral analyzers to classify a host based on its behaviors. Correlation strategies are employed on the aggregated behaviors and behavioral score to cluster hosts. The overall network object trust uses a maximum likelihood classification mechanism over the results from behavioral trust, classification, and correlation results to produce the overall trust of the object.

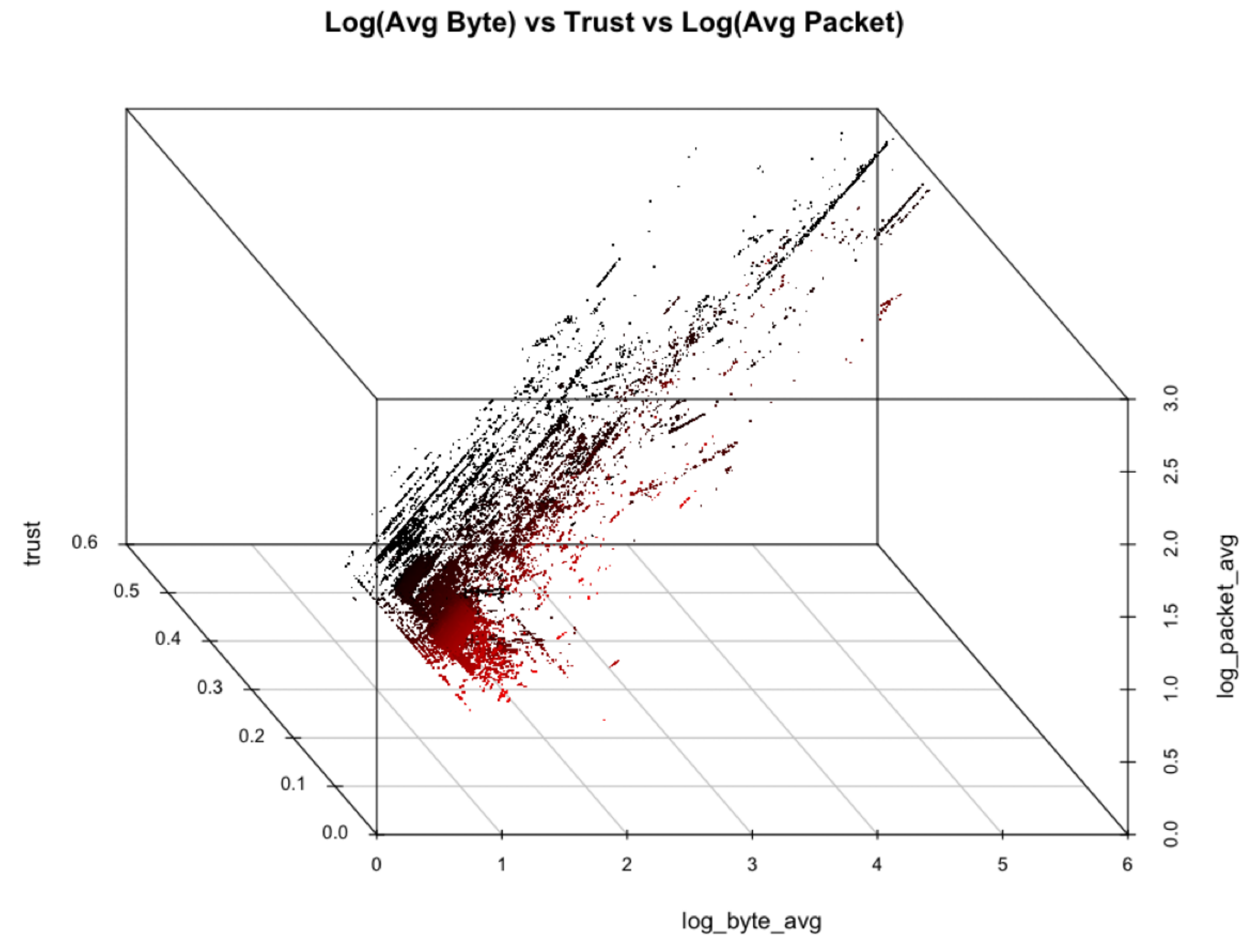

Figure 5: Trust visualized with network behaviors

4.2 Trust

Computational trust provides a metric that can be used to derive risk from the raw data being analyzed [12], and is calculated by the weighted summation of behavioral trust scores derived from the behavioral analysis functions. This behavioral trust is driven by the security policies employed within an agency e.g. NATO.

Object trust is then a function of the summation of all of the behavioral trusts, where the trust model weights each behavioral trust score. In this model, normal and behavioral anomalies are represented using a number of techniques, including clustering and support vector machines. Once these behavioral patterns have been established, they can be used to drive the trust score of network objects being followed by the fusion engine.

The results of our approach have provided qualitative visualizations and quantitative metrics associated with botnet detection using a hyperplane approach [20]. We have applied the system to live data and found weak signal data exfiltration behaviors (not detected by conventional tools) by leveraging regional aspects of the IP address space (Figure 4). We found significant bidirectional UDP traffic after discovering outgoing flows to China via port 9000.

Some of the views into the feature space provide insight into normal network usage (Figure 1). We performed network forensics on a data set and found the use of file transfer being done on a compromised machine (Figure 4). By using trust as a metric, the system is able to provide an abstract means through which threats can be identified and information can be shared between various sensors and detection systems. The trust model allows for the measurement of trust through a set of concurrent behavioral analysis strategies. Trust measurements can help the analyst focus attention on threats that are important to specific needs, and reduce the inundation of meaningless alerts and data.

5.0 CONCLUSIONS

It is this fundamental ubiquitous nature of our future networks, like NAF, that should drive our notions of trust creating technology and operations that provide end-to-end security. Such needs push past single vender solutions requirements standards to drive the development of open and layered architectures supporting trustworthiness.

Objective trust, defined objectively through a cryptographic means, provides a core component of trust focused on the management of identity. Identity alone does not provide the detail of trust needed today's ubiquitous and dynamic systems. A layer of behavioral trust is needed to augment the trustworthiness of devices within our enterprises, providing a subjective view into trust driven by the security policies of a distributed and complex network.

In 2009, NIST stated that "numerous problems have been identified in current key management methodologies" [5]. In this paper we have explored how NIST's call for IdM/CKM designers to move away from public key technologies suggests a possible re-emergence of online IdM/CKM systems and the opportunity to integrate behavioural trust with objective trust.

Biological Immune Systems concepts have been applied to intrusion detection systems [24]. In this paper we focused on the establishment of digital self non-self for the enterprise being driven by trust as a core metric. That is does a ubiquitous system trust the identify of a device, or that devices behavior within the network environment.

We defined trust as a layered concept that moves from objective trust defined as identify of the object to subjective trust defined by the objects behaviors. We highlighted the challenge in creating a distributed concept of trust that is a combination of multiple dynamic layers rooted in both objective and subjective models. We briefly mentioned Synaptic Laboratory's global IdM/CKM proposal which distributes the execution of provisioned services across m autonomously owned/managed service providers to mitigate insider fraud/attacks. We propose that behavioral trust derived from network behaviors can be layered within the trusted provenance providing a complementary addition to a trusted provenance.

Our contribution in the evolution of computational trust is the need to formalize a model that can be layered with existing concepts of identity management. Global identity management is both "objective", and "subjective". For example, a passport from a third world country may make an objective assertion, however as a UK/AU/US Government, do I trust that objective assertion is correct? This is where we perhaps use multiple-attested identities to increase our confidence on the static assertions that we are validating. We view behaviorally derived trust as being subjective and managed locally based on the security policies within the environment. This is in contrast to notions of global identity management which is rooted in a more objective way. Another contribution we make in the area of behavioral analysis and trust is distinguishing behavioral analysis from profiling. We propose that profiling implies analysis of an identity, where behavioral analysis is done independent of identity. Thus, behavioral analysis, when done without identity, enables the sharing of actionable information without infringing upon the privacy of individuals or the community. This can someday foster system-level collaboration of threat behaviors when other enabling technologies bridge the semantic gap allowing systems to share meaningful information and knowledge.

5.1 Future Work

A formal notion of a network behavior can facilitate systematic description of threats, and the sharing of actionable information between systems. Each behavioral feature can be represented as a primitive, what we define as a neteme, used to assemble a narrative used to identity threats.

An example of the use of primitives for threat detection can be used to discover bots using a distributed defense strategy. For example, a single bot in a botnet has a known lifecycle that can be broken down into [34]:

1. Establish C2; *e.g.*, P2P, Master/Slave, Hybrid and Layered, encrypted traffic,
2. Scan for vulnerable machines:,
3. Compromise the machine,
4. Recovery and configuration of compromised host; *e.g.*, disable virus protection software.
5. Reinforcement with new bot code after compromise,
6. Idle; *e.g.*, until notification, or with time-bomb.

Each phase in the lifecycle can be represented by sets of behavioral primitives used to classify the botnet of long time intervals.

This style of classification has been applied to Optical Character Recognition (OCR) in the early 1990's with the omni-font character recognition systems based on a structural classifiers moving away from pixel matching. These types of classifiers decomposed characters into simpler components, called primitives, and then arranging these primitives using various topological attributes [35] . Essentially, combining the primitives into a structural narrative of the character itself; *e.g.*, the character 'P' is made of a vertical line, and an arc connecting on the right side of the line. Aggregated behavioral analysis provides the same constructs as omni-font character recognition systems, differentiating itself from previous classification methodologies based on structural aspects of malware. Our approach treats behaviors as netemes, and uses them to define a pattern, or threat, as omni-font would a letter.

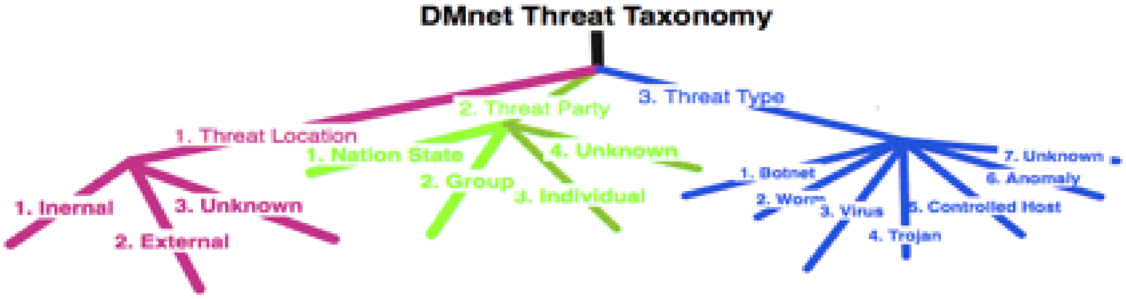

Figure 6: Notional Threat Ontology Promoting the Establishment of Behavioral Primitives

Another area of work lies in the establishment of a common semantic understanding of network behaviors, leading to the development of a behavioral Ontology. There has been much attention given to vulnerability analysis and the establishment of Common Vulnerability Exposure (CVE). We can perhaps flip this viewpoint and focus more on the source of the threat, rather on the sink, enabling the creation of technologies that formalize the descriptions of behaviors used to establish of common vocabulary through which systems share behavioral descriptions of threats; *e.g.*, narratives, in real-time.

We see an exciting new field opening up in the area of online global IdM/CKM architectures that support policy driven behavioural assertions for internal enterprise, business to business and global communities. Synaptic anticipates that the core IdM/CKM proposal can be finalised and deployed within a few years with an appropriate level of support. Our excitement in this field is shared by others. For example, Synaptic was invited to the 'closed' US NITRD National Cyber Leap Year Summit (August 2009) and several proposals related to Synaptic's global IdM/CKM were accepted and taken forward. Also, there are already approximately twenty corporations including NATO contractors, interested to collaborate in the development of this system.

ACKNOWLEDGMENTS

The authors want to acknowledge support from of the Cyber Security Program Area of the Command, Control and Interoperability Division within the Science and Technology Directorate of the U.S. Department of Homeland Security, especially the support from Dr. Douglas Maughan. We would also like to thank both Ron Kelson and Benjamin Gittins of Synaptic Laboratories Ltd [4] for a thorough review of our paper, providing a background for CKM and helping us to explore behavioural trust and identity.

REFERENCES

[1] NATO, "NATO Architecture Framework," Tech. rep., AC/322-D(2007)0048, 2007.

[2] USOWH, "Cyberspace Policy Review: Assuring a Trusted and Resilient Information and Communications Infrastructure (May 26, 2009)," United States, Office of the White House, May 2009, Available at `http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf`.

[3] CoChairs, N. C. L. Y., "National Cyber Leap Year Report Documents," Tech. rep., NITRD, 2009.

[4] Haldar, V. and Franz, M., "Symmetric behavior-based trust: a new paradigm for internet computing," *NSPW '04: Proceedings of the 2004 workshop on New security paradigms*, ACM, New York, NY, USA, 2004, pp. 79–84.

[5] Barker, E., Branstad, D., Chokhani, S., and Smid, M., "Cryptographic Key Management Workshop Summary (draft)," Interagency Report 7609, NIST, June 2009.

[6] Gittins, B. and Kelson, R., "Overview of SLL's proposal in response to NIST's call for new global IdM/CKM designs without PKC: slideshow," *IEEE Key Management Summit 2010*, IEEE, Lake Tahoe, Nevada on May 4-5, 2010., May 2010, (To Appear).

[7] Weth, C. V. D. and Bhm, K., "A Unifying Framework for Behavior-based Trust Models," 2006.

[8] Denning, D. E., "An Intrusion-Detection Model," *IEEE Symposium on Security and Privacy*, 1986, pp. 118–133.

[4] `http://synaptic-labs.com/`

[9] Snow, B., “We Need Assurance!” *ACSAC ’05: Proceedings of the 21st Annual Computer Security Applications Conference*, IEEE Computer Society, Washington, DC, USA, Dec. 2005, pp. 3–10, Available at `http://www.acsac.org/2005/papers/Snow.pdf`, `http://ic.epfl.ch/webdav/site/ic/shared/ResearchDay/SnowSlides.pdf`.

[10] Rehak, M., Pechoucek, M., Grill, M., Stiborek, J., Barto, K., and Celeda, P., “Adaptive Multiagent System for Network Traffic Monitoring,” *IEEE Intelligent Systems*, Vol. 24, No. 3, 2009, pp. 16–25.

[11] Lim, S. Y. and Jones, A., “Network Anomaly Detection System: The State of Art of Network Behaviour Analysis,” 2008, pp. 459–465.

[12] Marsh, S. P., “Formalising trust as a computational concept,” Tech. rep., University of Stirling, 1994.

[13] “TrustedSource$^{\text{TM}}$,” Website., McAfee, 2010, http://www.trustedsource.org/.

[14] Cheney, J., Chong, S., Foster, N., Seltzer, M., and Vansummeren, S., “Provenance: a future history,” *OOPSLA ’09: Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, ACM, New York, NY, USA, 2009, pp. 957–964.

[15] Gray, E., marc Seigneur, J., Chen, Y., and Jensen, C., “Trust Propagation in Small Worlds,” *In Proc. of 1st Int. Conf. on Trust Management (iTrust03*, 2003, pp. 239–254.

[16] Lyle, J. and Martin, A., “Trusted Computing and Provenance: Better Together,” *TAPP 2010 Workshop on the Theory and Practice of Provenance*, 2010.

[17] Paxson, V., “Bro: a system for detecting network intruders in real-time,” *Computer Networks*, Vol. 31, No. 23-24, 1999, pp. 2435–2463.

[18] Roesch, M., “Snort: Lightweight Intrusion Detection for Networks,” *LISA*, 1999, pp. 229–238.

[19] Wang, K., Cretu, G. F., and Stolfo, S. J., “Anomalous Payload-Based Worm Detection and Signature Generation,” *RAID*, 2005, pp. 227–246.

[20] McCusker, O., Kiayias, A., Walluck, D., and Neumann, J., “A Combined Fusion and Mining Strategy for Detecting Botnets,” *CATCH ’09: Proceedings of the 2009 Cybersecurity Applications and Technologies Conference for Homeland Security*, 2009.

[21] Soldo, F., Le, A., and Markopoulou, A., “Predictive Blacklisting as an Implicit Recommendation System,” 2009.

[22] Forrest, S. and Longstaff, T. A., “A Sense of Self for Unix Processes,” *IEEE Symposium on Security and Privacy*, 1996.

[23] Forrest, S., Alan S, P., Allen, L., and Cherukuri, R., “Self-Nonself Discrimination in a Computer,” *IEEE Symposium on Security and Privacy*, 1994.

[24] Dasgupta, D., “Immuno-inspired autonomic system for cyber defense,” *information security technical report 12*, 2007.

[25] Bretscher, P. and Cohn, M., “A theory of self-nonself discrimination,” *Science*, 1970, pp. 1042–1049.

[26] Collins, M., “Flow Traffic Analysis Narratives,” 2010, http://www.cert.org/flocon/2010/flocon2010_abstracts.pdf.

[27] Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D. A., and de Weger, B. M. M., “Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate,” *CRYPTO ’09*, Vol. 5677 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, Aug. 2009, pp. 55–69.

[28] Gutmann, P., “Everything you Never Wanted to Know about PKI but were Forced to Find Out,” Nov. 2002, Available at http://www.cs.auckland.ac.nz/˜pgut001/pubs/pkitutorial.pdf.

[29] Gutmann, P., *Engineering Security*, (draft book), Dec. 2009, Available at http://www.cs.auckland.ac.nz/ pgut001/pubs/book.pdf.

[30] KPMG, “Profile of a Fraudster Survey 2007,” Forensic advisory, KPMG International, Apr. 2007, Available at http://www.kpmg.co.uk/pubs/ProfileofaFraudsterSurvey(web).pdf.

[31] Diffie, W. and Hellman, M. E., “Multiuser cryptographic techniques,” *AFIPS ’76*, ACM, New York, NY, USA, June 1976, pp. 109–112.

[32] de Secondat, Charles, B. d. M., *The Spirit of the Laws*, Crowder, Wark, and Payne, 1777.

[33] Tadda, G., Salerno, J. J., Boulware, D., Hinman, M., and Gorton, S., “Realizing situation awareness within a cyber environment,” *Proceedings of SPIE*, 2006.

[34] Liu, J., Xiao, Y., Ghaboosi, K., Deng, H., and Zhang, J., “Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures,” *EURASIP Journal on Wireless Communications and Networking*, 2009.

[35] Foggia, P., Sansone, C., Tortorella, F., and Vento, M., “Combining Statistical and Structural Approaches for Handwritten Character Description,” 1999.