2013 5th International Conference on Cyber Conflict
K. Podins, J. Stinissen, M. Maybaum (Eds.)

Deriving Behavior Primitives from Aggregate Network Features using Support Vector Machines

Owen McCusker
Guardian Services
Sonalysts, Inc
Waterford, CT, USA
mccusker@sonalysts.com

Scott Brunza
Guardian Services
Sonalysts, Inc
Waterford, CT, USA
scottso@sonalysts.com

Dipankar Dasgupta
Professor of Computer Science
University of Memphis
Memphis, TN, USA
dasgupta@memphis.edu

Abstract: Establishing long-view situation awareness of threat agents requires an operational capability that scales to large volumes of network data, leveraging the past to make-sense of the present and to anticipate the future. Yet, today we are dominated by short-view capabilities driven by misuse based strategies; triggered by the structural qualities of attack vectors. The structural aspects of cyber threats are in a constant flux, rendering most defensive technologies reactive to previously unknown attack vectors. Unlike structural signature based approaches, both the real-time and aggregate behaviors exhibited by cyber threats over a network provide insight into making-sense of anomalies found on our networks. In this work, we explore the challenges posed in identifying and developing a set of behavior primitives that facilitate the creation of threat narratives use to describe cyber threats anomalies. Thus, we investigate the use aggregate behaviors derived from network flow data establishing initial behavior models used to detect complex cyber threats such as Advanced Persistent Threats (APTs). Our cyber data fusion prototype employs a unique layered methodology that extracts features from network flow data aggregating it by time. This approach is more scalable and flexible in its application in large network data volumes. The preliminary evaluation of the proposed methodology and supporting models shows some promising results.

Keywords: *Behavior analysis, aggregate behaviors, network flow analysis, anomaly detection, machine learning*

1. INTRODUCTION

The North Atlantic Treaty Organization (NATO) is faced with the increasing need to support international operations that leverage the use of complex end-to-end 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. The attack surfaces associated with such systems allows cyber threat agents (e.g. nation states, hackivists) to employ the use many types of coupled attack vectors, such as phishing and key-logging; gaining access and persisting in these environments for years unnoticed. The complexity of these cyber threat agents has grown steadily in recent years, and is exhibited in the employment of distributed cyber missions that operate over various time scales within our Information and Communication Technology (ICT).

In 2013, Kaspersky Lab uncovered the actions of ‘’Red October” which they feel has been harvesting intelligence from high profile organizations since 2007 [2]. This espionage group incorporated a set of simple attack vectors that allowed them to penetrate and persist in both public and private organizations for prolonged period of time. According to Verizon in 2012, threat agents incorporate multiple threat actions during an attack, and these attacks can go on form months well within our supply chains and distributed throughout our networks [3]. Yet our detection models and cyber defence capabilities are still tuned for single ingress points, and mostly employ rule-based defensive strategies.

There is an array of defence-in-depth capabilities that can be employed in concert to deter the sophisticated attacks from threat agents including: firewalls, Multi-factor authentication, role and attribute based access control end-point security, Network Intrusion Detection/Prevention Systems (IDS/IPS) training and policy creation and enforcement. Each capability provides deterrence to attack vectors in a slightly different way. Most monitoring and response capabilities can be categorized into misuse detection and anomaly detection. While the misuse detection can only detect known attacks, anomaly detection on the other hand can detect unknown and zero-day attacks. However, anomaly-based detection methods suffer from false positives, as all anomalies may not relate to attacks. This work is on leveraging behaviour-based anomaly detection with focus on hierarchical aggregated features/ attributes of monitored hosts.

One of the reasons why threat agents pose such a significant risk to national infrastructure is that cyber defence capabilities are dominated by misuse-based capabilities that provide short-view situation awareness. Most of these capabilities

correlate volumes of real-time events making sense of what is happening at any given instant in time but do not scale well over longer time periods. The anomaly detection paradigm offers the ability to adapt to emergent threats based on past events.

In of 2009, BBN addressed this issue by proposing a notional architecture that can scale at increasing network speeds using event aggregation [4]. Key to the success in their approach is the use of Scyllarus, an event correlation system [5]. This correlation system clusters events by measuring the similarity of their attributes. Our position is to take a host-centric posture, instead of event-centric, focusing on the aggregate behaviours of hosts as extracted from using network flow traffic.

Over the past few years behaviour-based models have emerged to bridge the gap in capability focused on anomaly defection of emergent threats [6], [7], [8]. These systems are mostly event-centric, where behaviours are extracted from event features and aggregated over time in terms of a source and a destination. For example, in [7], aggregate event graphs are used make sense of behaviours obtained from sensors. The event takes into account both the source and destination providing a connection, or edge in the graph. In another example, Rehak uses classifiers agents to score events as legitimate or malicious [6]. Lastly, LNLL created a system SETAC that uses a distributed model to detect both local and global anomalistic behaviours within their networks [8]. Unlike the previous systems, we position our host-centric work to develop layers of classifiers, with the first intermediate step toward establishing a set of primitives used measure overall behaviours of hosts.

In our previous work [9], we developed a host-centric cyber data fusion capability based on a layered methodology (Figure 1), which transforms network flow data into aggregate features of hosts over various time windows. We collected network flow data using SiLK over a period of six months to begin our exploration of the data [10]. One of our findings suggests that when a group of host is observed over a period of time, they behave in very consistent ways.

In our current work, we are looking to develop an adaptive methodology that leverages the past aggregate behaviours of normal operation of systems to build a predictive model. We then compare the predicted behaviour of a host or set of hosts with the actual behaviours to determine the classification of the abnormality of a host. The overall classification is measured in terms of a set of behaviour primitives. In order to minimize false positives in attack detection, this approach incorporates some signalling mechanism similar to the biological immune system.

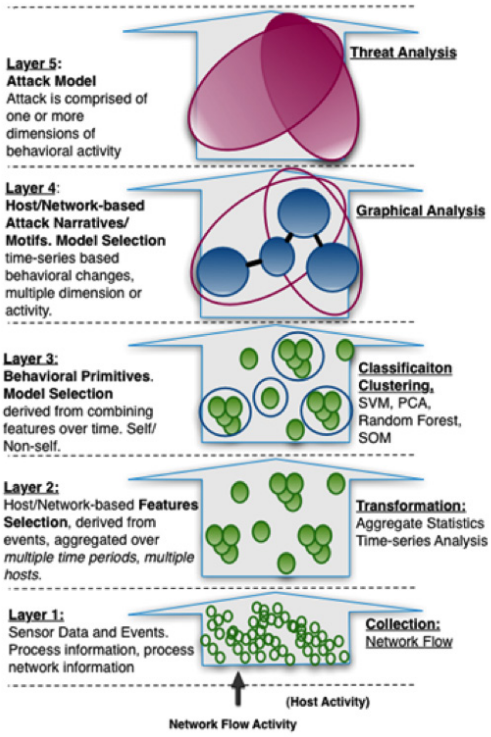

Figure 1. A Layered Methdology. Most CND technology is developed to operate in Layer 1, processing volumes of raw data and events from sensor technology. Our unique approach transforms the data first, Layer 2, and then applies classification models in Layer 3 and above

Figure 1 illustrates a layered detection methodology approach, which allows the Team to work independently at different abstraction layers of the overall problem. In this approach, we can focus on developing algorithms in Layer 1-3 and in the future we can focus on social-based algorithms in Layer 4-5. We envision multiple algorithms to leverage in the layered model.

The approach we take in this research is to establish a rich set of behaviour primitives that facilitates long-view situation awareness. The behaviour primitives represent the basis for a behavioural language through which we can someday create threat narratives that are shared as actionable intelligence in real-time throughout a trusted community of cyber defenders. Narratives are viewed as graphs of behaviour primitives that capture aggregate description of threat agents. These threat narratives can represent social relationships and/or characteristics that are shared between a group of hosts, geographic region, and/or autonomous system.

This paper is organized as follows. In Section 2 we review work related our proposed approach. Section 3 we discuss our methodology [11], which includes ground truth development, feature selection, model development and model evaluation. In Section 4 we discuss the conclusions of our results.

2. RELATED WORK

In this section, we review related intrusion detection research leading to behavior analysis. We then present important works on distributed collaboration and correlation, intrusion detection leveraging network flow, cyber situation awareness, and review pertinent work on knowledge discovery. This section contrasts the evolving threat with the models that were used in establishing existing detection technologies.

In 1987, an intrusion detection model proposed by Denning focused on the identification of network attacks directed toward a single host [12]. The threat, at that time, was comprised mostly of attackers attempting to gain remote access to a host. Soon after this model was proposed, the introduction of worms was officially acknowledged by the release of the Morris Worm in 1988 by Robert T. Morris [13]. Since this time, there has been a constant tug-of-war between the introduction of new threat types and the development of new techniques to meet the evolving detection requirements. Ghosh et al. [11] developed an application level behavior model for intrusion detection.

A. *MULTI-EVENT CORRELATION AND DISTRIBUTED COLLABORATION*

In Section 1, as discussed by [4], event correlation can facilitate aggregation and scaling to network speed. BotHunter [14] is a system built specifically for the correlation of events occurring within specific network locales. This system focuses on detecting network dialog communications between various bots within a botnet and is driven by alerts from SNORT [15]. These dialogs represent different communication behaviors exhibited by a bot during its lifecycle. An event trail is created that triggers an alert based on specific bot behaviors that occur.

The Worminator project leverages the distributed collaboration of events generated from an IDS in order to establish attack patterns [16]. The system leverages alert aggregation and reduction to reduce the cost of the exchanging raw data. A correlation scheduler is used to set up peers to exchange alerts. The Worminator paper highlights the need to reduce and manage the large volumes of alerts that are exchanged between detection peers. Worminator uses Bloom filters to manage privacy by setting up private watch lists.

Our proposed host-centric model is driven by network flow captured using SiLK instead of an event-centric IDS. We derive profiles consisting of features extracted from the network communication between various hosts. These behavior profiles are fed into a classification and correlation engine.

B. KNOWLEDGE DISCOVERY AND ADAPTABILITY

In Section 1 we discussed the knowledge discovery needs for a system to adapt by leveraging the past to the present in [6]. Knowledge engineering has been applied to intrusion detection in MADAM ID [17] where association rules mining was used offline to construct new rules to detect threats in a misuse detection system. Knowledge discovery has been applied in another way for misuse detection in the Intelligent Intrusion Detection System (IIDS) [18]. Misuse signatures are viewed as rules through which a genetic algorithm creates a set of rules by combining behaviors based on network connection information. In both cases, rules are directly related to threat signatures. We propose a more abstract view dealing with knowledge discovery, where threats are represented in a set of behavior primitives and extracted features.

C. DMNET – A CYBER DATA FUSION PROTOTYPE

The overall system [9] focuses on the notion of tracking various network objects, O, e.g. hosts, hostgroups, and networks, and determining if they are threats. Tracking these objects involves collecting events and data from a number of different network sensors, e.g., network flow, NIDS, honeypots, and creating a sample space.

In our current data fusion system, network flow data and alerts generated by network sensors reflect the totality of information and model's sample space, S, available to the detection system regarding the objects to be analyzed.

To utilize this data, it is first normalized and transformed into a representation that is conducive to algorithmic processing. The fusion engine operates over a sample space denoted as S representing sensor data. This fusion operation is represented by an object behavioral analysis function, B.

The aggregated behavioral analysis of the sample for a specific object O, $B(S_O)$, produces a feature characteristic, or behavior, for that object denoted by F_0 accumulated within a set time window $F_{tw,O}$. The sample space, S, is then transformed into an aggregated feature space F_S. The Time window, tw, consists of periods such as hour, day, month, year. F_0 is represented by a n-tuple, or n-gram, of individual time-based features, for example $F_{month,O} = \langle f_1, f_2, \ldots f_n \rangle$, describes O over a period of a month. These features consist of structural, behavioral, and/or application specific properties of O over a given time period.

Information from the deployed sensors is fed into the fusion engine. Sensors could include a variety of network, appliance, or host-based software or hardware. The sensor information could be in the form of netflow [10] or pcap records, network intrusion detection/prevention system feeds, alerts from honeypots, or anti-virus reports. This information is parsed and then normalized by a perception module. Normalization refers to the process of converting the parsed information into a form that is standard and readily understood and manipulated by modules further down in the processing chain.

The data fusion component maps normalized data to vectors of high dimensionality. This is achieved by a profiling function that parses the raw normalized events produced by the vectors and aggregates them to form a basic network object and embeds them in a vector space. After the profiling is completed, each fusion element is associated with a feature characteristic that describes it according to the profiling function that was applied. Note that the features that can be extracted depend upon the type of sensor provided to the system as a source of network data. They range from summary data such as netflow, to fine-grained information such as pcap header dumps produced by tcpdump.

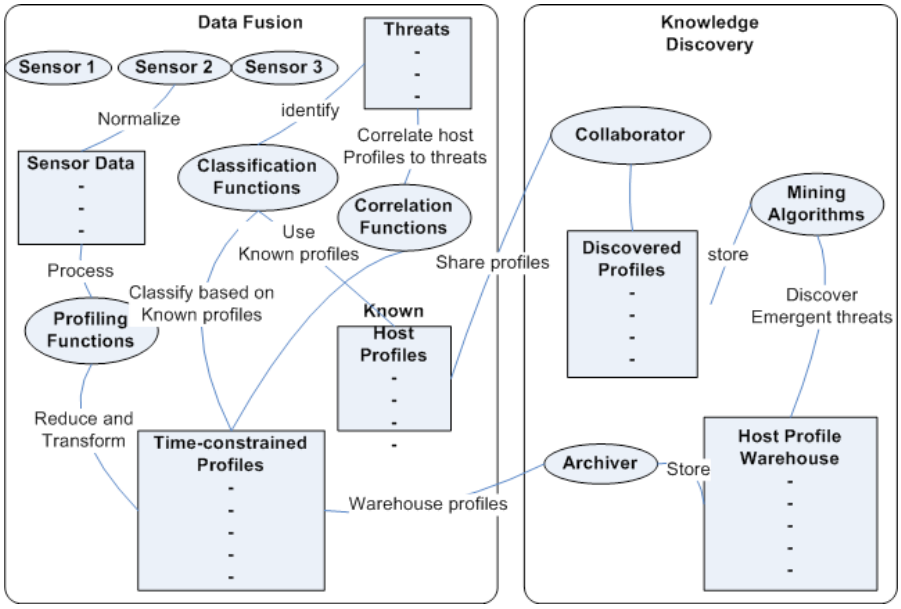

Figure 2. Dmnet Cyber Data Fusion Prototype. This architecture represents a combined fusion and data mining methdology.

D. AGGREGATE BEHAVIOR ANALYSIS

Most current technology operates at Layer 1 (Figure 1) in our methodology applying classification models to raw data and sensor events. We need technologies that scales to the volumes of data and events being created by our cyber sensors.

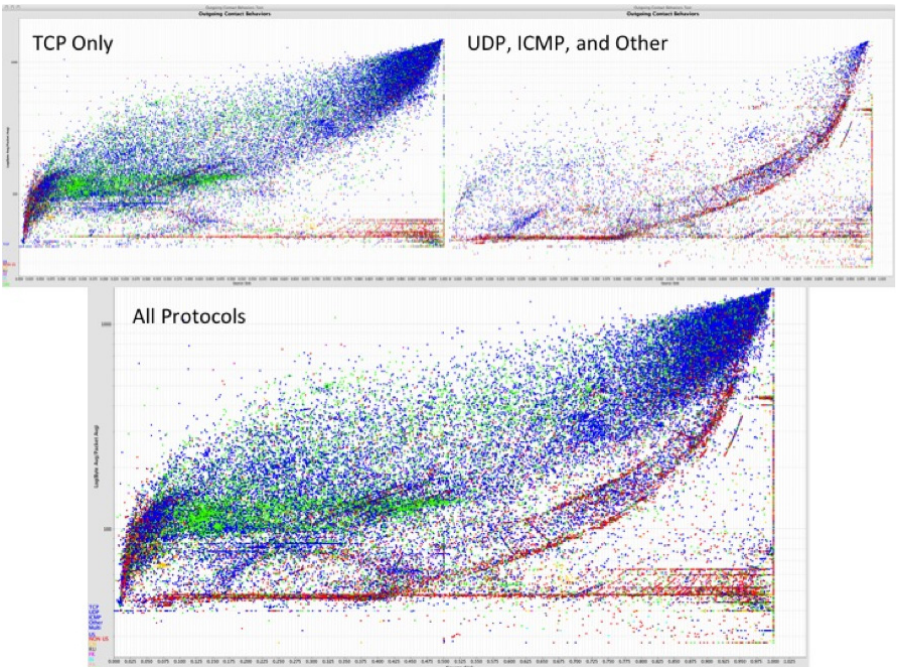

Figure 3. Behavioral Visualization Created From Data Generated by Fusion Prototype. There are three different visualizations depicted showing UDP behaviors (top right), TCP Behaviors (top left) and All protocols bottom. Each dot represents a host in a behavioral feature space. These diagrams show the behaviors of hosts going from "source to sink behaviors," where hosts receiving data from our system are to the left, and hosts sending data to our system are to the right.

In previous years, Sonalysts started the development of a disruptive cyber fusion approach based on aggregate behavioral analysis. Our approach transforms this data, Layer 2, first into a rich multivariate features space before we apply classification models (Layer 3).

Layer 1 CND technologies cannot scale well when faced with the increasing amount of network traffic. By transforming this raw data into behaviors we can aggregate it into multiple time periods and provide a data reduction technique that can begin to scale to the increase in traffic volumes.

3. CHARACTERIZING BEHAVIOR PRIMITIVES

This section highlights the overall methodology for model development that is being employed to detect behavioral primitives enumerated in the ground truth data set. We evaluate the feasibility of our methodology by applying to three different types of classification models focused on the identification of pinging, or beacon-like behaviors. This is one of many types of behavioral primitives that we are working on quantifying as part of the ongoing research.

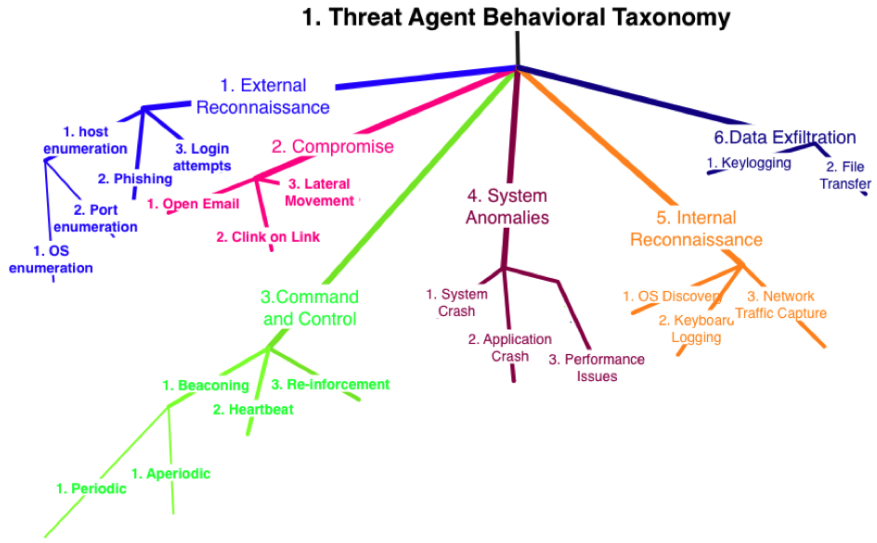

Figure 4. Behavior Primitive Taxonomy

The behavioral primitives are captured in a behavioral taxonomy (as shown in Figure 4). Each node represents a primitive that can be measured in terms of a set of features created by our fusion engine and by a classification model. For example, in this paper beaconing behavior is modelled using a Support Vector Machine (SVM) in terms of three aggregate features: outgoing work, outgoing byte variance, and source sink. Outgoing work is defined as the average bytes per packet that leaving a network device e.g. host. Outgoing byte variance is measures the changes in bytes per packet in outgoing flow traffic from hosts. Source sink is a measure of the directionality of traffic from network device and has a value of 0 to 1. Where purely beaconing devices have a value of 1.

A. GROUND TRUTH

The research is leveraging ground truth behavioral data gathered between the months of December 2010 and to February 2011. This data is derived from live network flow traffic that we continually capture on our networks and transform into a behavioral features space. The goal in leveraging this ground truth is produce a set of behavioral primitives that can be used to perform predictive analytics using a number of learned models.

The behavioral data for the work is gathered from a number of discrete vantage points: External to the firewall focused on non-assets hosts (not managed by the client), internal focused on non-assets hosts, and internal vantage point focused on assets. We have been gathering behavior data actively since 2009 and to date we have shared ground truth data with institutions to promote aggregate behavior analysis (2010, Oakridge National Laboratory[1].)

1) Meaningful Indicators

We have identified a number of meaningful indicators during the analysis of the three ground truth data sets. Some of these indicators are highlighted in this action of the document.

a) External Vantage Point Non-Assets

There are over 1.7 million hosts being followed in the external vantage ground truth data set. The data set is rich with host behaviors found in both monthly and daily time aggregates. The following picture highlights abnormal activity, against policy of a host running a Unreal Tournament client and having it beacon out to a number of external server hosts. This asset is compromised a few weeks later.

b) Internal Vantage Point Non-Assets

The internal vantage point provides insight to actual communications between assets and non-assets, without the noise inherent from outside the firewall. In (Figure 5) there is a cluster of behaviors associated with internal hosts performing a heartbeat out to Japan. There are multiple machines that are sending a consistent amount of bytes and packets to this server. These machines are on a internal subnet through which there where known compromised machines.

---

[1] Oakridge National Laboratory, Computational Intelligence Behavior Modeling Laboratory, promoting the use of scalable algorithm development using High Performance Computing technologies, http://csiir.ornl.gov/

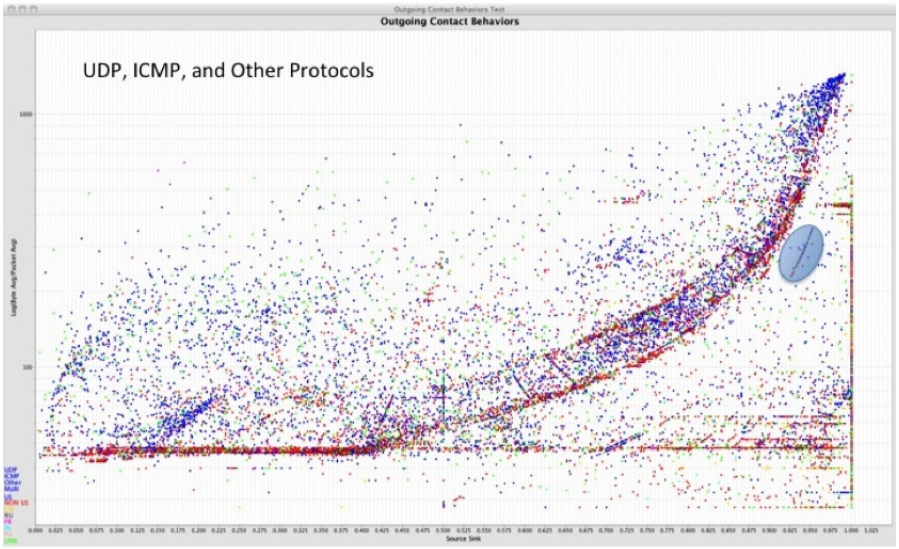

Figure 5. Visualization of Monthly External (non-assets) Host behaviors from the External Vantage Point. All traffic in this view is filtered out except for UDP, ICMP, and Other. See Figure 3 for a comparison between the protocol specific behaviors. The highlighted points are that of a single host in mid December launching Unreal Tournament and beaconing to sites around the globe

c) Internal Vantage Point Assets

The Internal vantage point ground truth data set offers the highest fidelity of behavioral features. We are only tracking 1400 hosts from this vantage point compared to 1.7M hosts on the external one. Having a smaller amount of contacts can allow us to focus on finer grained temporal features looking into both the quantification of normal and abnormal behaviors that provide a side-by-side comparison of host behaviors looking at byte and packet usage.

B. BEHAVIOR TAXONOMY

In a paper delivered to NATO in 2010 (and based on our work for DHS S&T from 2006), we established two taxonomies facilitating the understanding of trust in end-to-end systems [19]: sensor taxonomy, and a behavioral taxonomy. The sensor taxonomy provides a basis for which we associate what behavioral features are derived from the various sensors employed by the system. We are further refining the two sets of taxonomies to support our work. In the future, these taxonomies will be developed into feature Ontologies with the addition of meaningful attributes to each node. The goal of this work is to identify behavioral primitives derived from the analysis of sensor data.

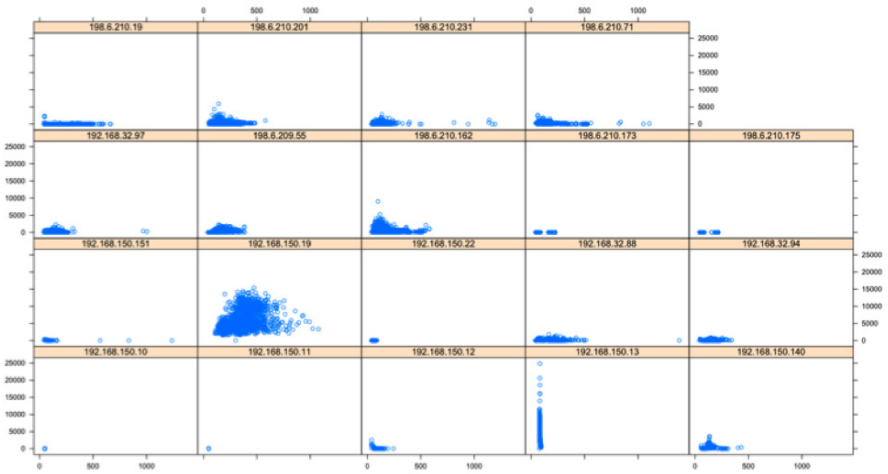

Figure 6. *Comparative Behaviors of Internal Hosts (Assets).* In this graph we are looking at byte versus packet behaviors. This side-by-side visualization of internal asset behaviors presents the degree of behavioral differences between hosts. The host in row 3, from the top, and column 2 from the left is the email server. The host in row 4 and column 4 is a DNS

To date, we are only focused on network flow. In the future, we plan on integrating other types of sensor technology into the prototype. The prototype already has an extensible sensor management framework.

We expect this development to be iterative in nature and mature as we begin to apply multiple classification models to derive meaningful behavioral primitives.

The behavior taxonomy (Figure 4) serves as a way to organize the various behavioral primitives that are being researched within the ground truth data set. Our goal is to have a way to score each of the behavioral primitives found within the taxonomy based on a specific classification model. A first attempt in modeling behavioral primitives is focused on beaconing behaviors as addressed in the previous section. We will plan on choosing multiple features for each model. Our goal is to be able to correlate multiple behavioral features to create graphical narrative describing threat agent behaviors.

C. BEHAVIOR MODEL DEVELOPMENT

Our techniques differ for a number of approaches that focus on the detection of specific classes of applications and attacks using models such as SVMs. Instead of classifying each individual flow of communication from a host we focus on the aggregation of transformed features to one specific host. By taking a host-

centric approach in our methodology we are able to collect meaningful behavioral aggregations of hosts, subnets, and geographic regions.

Li *et al.* [20] have applied the use of SVMs to detect seven classes of applications with optimized yields of 96.4% accuracy with un-biased training data. Their work has classified the following types of applications: Bulk (ftp), interactive (ssh, telnet, rlogin), mail (pop, smtp, imap), service (x11, dns), www (http, https), p2p (kazaa, bittorrent, gnutella), multimedia (voice, video streaming), game (half-life), attack (worms, virus), and other. The approach, although accurate, is high grained.

Instead of using a SVM to classify an application, our approach is finer grained in that by decomposing an application, or a threat agent, into a set of behaviors we will create behavioral language, or narrative, used to describe the threat actions over time. Lastly, instead of focusing on one specific model we are researching a number of models that operate over various time-based behavioral apertures or granularities.

1) Model Development using Support Vector Machine

Our initial goal is to focus on the predictive performance associated with ability to score behavioral primitives. There are a number of existing criteria that exists for evaluating models: predictive performance, interoperability, and computational efficiency. One reason for this choice is that our methodology allows for the concurrent processing of multiple models, which can be an area we focus on in future spirals. Our ultimate goals is to develop a set of primitives using supervised learning methods and then to augment this approach with unsupervised learning methods with the larger data sets. Essentially deriving new models, or variations of models, adding to our behavioral Ontology. For example, there can exist different variations of beaconing used by threat agents as they penetrate our systems. We will evaluate our models using receiver operator characteristic (ROC) curves.

2) Support Vector Machine Model Evaluation

A Support Vector Machines (SVM) represent a supervised pattern recognition algorithm used for binary classification problems. Being a supervised method, we are using our ground truth data set to train a SVM to detect various types of behavioral primitives, beaconing being the first. We are using the LibSVM library and R to apply SVM to our data set[2]. Unlike previous research done in our community [20], we are applying SVMs to host-centric behavioral features. Most of the research to date has applied these models to network communications and raw flow data. Within our methodology we have transformed the data into a

---

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

host-centric features space before we apply our models. Scaling the data input into the SVM is important. Without doing so the attributes with the higher numeric ranges can dominate the models output. This is especially true when using linear or polynomial kernels.

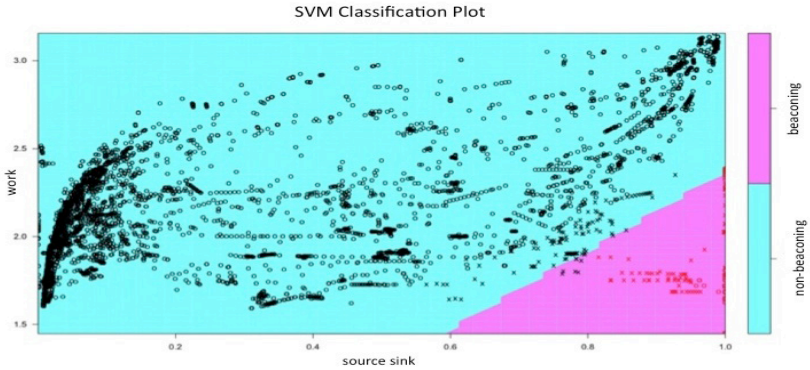

Figure 7. Trained SVM Visualization. Beacon behavior is below and to the right of the hyperplane

To assess Beaconing event (Figure 7) detection accuracy in a threshold-independent manner we use Receiver Operating Characteristic (ROC) curves (Figure 8), i.e., plots of achievable sensitivity vs. false positive rates, where the Sensitivity/True Positive Rate (TPR) is defined as the ratio between the number of Beaconing events (TP) flagged by the algorithm and the total number of Known Beaconing events (P), and the False Positive rate (FPR) is defined as the ratio between the number of non-Beaconing events (FP) flagged by the algorithm and the total number of non-Beaconing events (N).

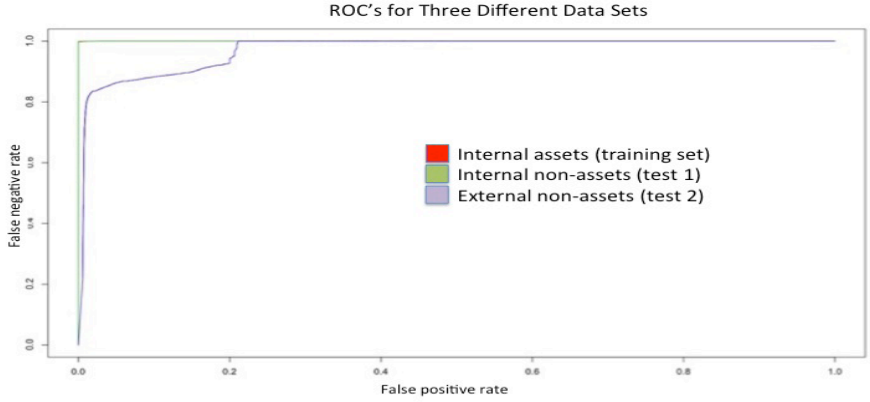

Figure 8. ROC Curve Results. Three different data sets were used: internal assets, internal non-assets, and external non-assets. Both the internal assets and internal non-assets exhibited a clean seperation between beaconing and non-beacon like behavior.

Our work focuses on two-class prediction problems (binary classification) where on class is a positive outcome and the other class has a negative outcome. We present contingency tables in the evaluation of a SVM in classifying pinging, or beacon-like behaviors.

Each point in this visualization (Figure 7) represents a days worth of host behaviors. The block circles are the non-pinging behaviors, and the red circles represent pinging behaviors. The light regions to the left and up are the predicted non-pinging behavior regions and the ping is the predicted pinging region. We are leveraging the use of SVMs to identify behaviors within the data set. We have selected work and source sink features to run against the model to detect beaconing behavior. We train the model with from our ground truth data sets focusing on pure beaconing behavior that exists, where source sink has a value of 1.

The training data set, taken from the internal vantage point, contained 6,635 hosts. The number of hosts exhibiting beaconing was 480. The following contingency table relates the true positive results to the predicted results and shows that 2 hosts where incorrectly predicted in the model. We labeled the data based on source sink and work feature values. This data is biased based on our labeling. We will run the data later on more unbiased data sets.

The unknown data set 1 contained 68,165 hosts. There were very few hosts having behavior indicative beaconing. The number of hosts exhibiting beaconing was 39 and had no false positive or negative errors in this data set.

The Test Data Set 2 (by see Table I) results show that our model was 86.9% accurate using the model developed from the internal training set. The contingency table provides an overview of the false positives (FP) 12,147 hosts, and false negatives (FN) of 50,031 hosts.

Table I. Contingency Table for Data Set 2

		Observed		
		0	1	Total
Predicted	0	195902	12147	208049
	1	50031	214583	264614
Total		226730	245933	472663

4. CONCLUSION

In this paper, we introduced a methodology for establishing behavior primitives in facilitating the creation of long-view situation awareness. Beaconing is just one primitive we will identity, and in our research are looking to grow that list of primitives to a few hundred

We discussed the concept of a behavior aggregation and its use in accurately measuring beaconing. In our work, the establishment of behavior primitives as an integral step leading to future detection, trust and risk models detecting and anticipating emergent behavior of compromised networked devices.

System behaviors can be used to develop models of trust to secure complex network [6], [19], [21], where trust is modeled from changes in past behaviors.

We have presented a classification model that utilizes aggregate features to create behavior profiles using a prototype cyber data fusion system. Since Denning proposed an alert-centric intrusion detection model back in 1987 protecting hosts from threats [12], new detection models are needed to advanced persistent threats (ATPs) that are realized from multiple ingress points within a network. The foundation of our work resides in the use of profiles in:

- The realization of behavior primitives to be later used in the knowledge discovery system,
- Collaboration between the discovery system and the fusion system, and
- The future establishment of threats in terms of behavior graphs in the fusion system.

Acknowledgment

Sonalysts would like 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.

REFERENCES

[1] NATO, “NATO Architecture Framework,” NATO, Technical Report 2007.

[2] Kaspersky. (2013, Jan.) www.securelist.com. [Online]. http://www.securelist.com/en/analysis/204792262/Red_October_Diplomatic_Cyber_Attacks_Investigation

[3] Verizon, “2012 Breach Investigation Report,” Verizon, Technical Report 2012.

[4] T. Strayer et al., “An Architecture for Scalable Network Defense,” BBN, Technical Report 2009.

[5] W. Heimerdinger, “Scyllarus intrusion detection report correlator and analyzer,” in *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, vol. 2, 2003, pp. 24-26.

[6] Martin Rehak et al., “Dynamic information source selection for intrusion detection systems,” in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, vol. 2, Richland, SC, 2009, pp. 1009-1016.

[7] B Czejdo, E Ferragut, J Goodall, and J Laska, “Network Intrusion Detection and Visualization Using Aggregations in a Cyber Security Data Warehouse,” *Int. J. Communications, Network and System Sciences*, vol. 5, pp. 593-602, Sept 2012.

[8] Arner Heller. (2010, Jan) str.llnl.gov. [Online]. https://str.llnl.gov/JanFeb10/matarazzo.html

[9] O McCusker, A. Kiayias, D. Walluck, and J. Neumann, “A Combined Fusion and Mining Strategy for Detecting Botnets,” in *ATCH ‘09: Proceedings of the 2009 Cybersecurity Applications and Technologies Conference for Homeland Security*, Washington, DC, 2009, pp. 273-284.

[10] Timothy Shimeall, Sidney Faber, Markus DeShon, and Andrew Kompanek. (2010, Jan) Using SiLK for Network Traffic Analysis. [Online]. http://tools.netsa.cert.org/silk/analysis-handbook.pdf

[11] Anup K. Ghosh, Aaron Schwartzbard, and Michael Schatz, “Learning Program Behavior Profiles for Intrusion Detection.,” in *In USENIX Proceedings of the Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, California, USA, April 9-12, 1999.

[12] Dorothy E. Denning, “An Intrusion Detection Model,” in *Symp. on Security and Privacy*, Feb 1986, pp. 118-133.

[13] Eugene H. Spafford. (1988, Dec) spaf.cerias.purdue.edu. [Online]. http://spaf.cerias.purdue.edu/tech-reps/823.pdf

[14] Guofei GU, Phillip Poras, Vinod Yegneswaran, Martin Fong, and Wenke Lee, “BotHunter: detecting malware infection through IDS-driven dialog correlation,” in *SS’07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, 2007, pp. 1-16.

[15] Martin Roesch, “Snort: Lightweight Intrusion Detection for Networks,” in *Proceedings of LISA ‘99: 13th Systems Administration Conference*, 1999, pp. 229-238.

[16] M.E. Locasto et al., “Collaborative Distributed Intrusion Detection,” Columbia University, Technical Report CUCS-012-04, 2004.

[17] Wenke Lee and Salvatore J. Stolfo, “Combining Knowledge Discovery and Knowledge Engineering to Build IDSs,” in *Recent Advances in Intrusion Detection*, 1999.

[18] We Li, “Using Genetic Algorithm for Network Intrusion Detection,” in *In Proc. United States Department of Energy Cyber Security Group 2004 Training Conference*, 2004, pp. 24-27.

[19] Owen McCusker et al., “Combining Trust and Behavioral Analysis to Detect Security Threats in Open Environments,” in *NATO/OTAN*, 2010, RTO-MP-IST-091.

[20] Zhu Li, Ruixi Yuan, and Xiaohong Guan, “Accurate Classification of the Internet Traffic Based on the SVM Method,” in *2007. ICC ‘07. IEEE International Conference on Communications*, 2007, pp. 1373 -1378.

[21] O McCusker, B Gittens, J. Glanfield, S. Brunza, and S. Brooks, “The Need to Consider Both Object Identity and Behavior in Establishing the Trustworthiness of Network Devices within a Smart Grid,” in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, vol. 54, 2010, pp. 1-4, 10.1145/1852666.1852724.

[22] Vern Paxson, “Bro: a system for detecting network intruders in real-time,” in *7th USENIX Security Symposium*, 1998.

[23] H. S. Javitz and A. Valdes, “The SRI IDES Statistical Anomaly Detector,” in *IEEE Symposium on Security and Privacy*, 1991, pp. 316-326.

[24] John McHugh, “Sets, Bags, and Rock and Roll: Analyzing Large Data Sets of Network Data,” in *ESORICS, 2004*, 2004, pp. 407-422.

[25] CERT. System for Internet Level Knowledge. [Online]. http://tools.netsa.cert.org/silk/

[26] Carrie Gates and John McHugh, “The Contact Surface: A Technique for Exploring Internet Scale Emergent Behaviors,” in *DIMVA, 2008*, 2008, pp. 228-246.

[27] Gerhard Munz and Georg Carle, “Real-time Analysis of Flow Data for Network Attack Detection,” in *Integrated Network Management*, 2007, pp. 100-108.

[28] CESNET, “Network Security Monitoring and Behavior Analysis: Best Practices Document,” CESNET, Technical Report 2011.

[29] Shu Yun Lim and Andy Jones, “Network Anomaly Detection System: The State of Art of Network Behaviour Analysis,” in *Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology*, 2008, pp. 459--465.

[30] Calvin Ko, “Execution monitoring of security-critical programs in distributed systems: A specification-based approach,” in *In Proceedings of the 1997 IEEE Symposium on Security and Privacy*, 1997, pp. 175--187.