Using Machine Learning for Behavior-Based Access Control: Scalable Anomaly Detection on TCP Connections and HTTP Requests

Aaron Adler*, Michael J. Mayhew†, Jeffrey Cleveland*, Michael Atighetchi*, and Rachel Greenstadt‡
*Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138
Telephone: (617) 873-3517, (617) 873-2515, (617) 873–1679
Email: {aadler, jcleveland, matighet} @bbn.com
† Air Force Research Laboratory, 525 Brooks Road, Rome, NY 13441
Telephone: (315) 330–2898
Email: Michael.Mayhew@rl.af.mil
‡ Department of Computer Science, Drexel University, Philadelphia, PA
Telephone: (215) 895–2920
Email: greenie@cs.drexel.edu

***Abstract*—Today's business processes are more connected than ever before, driven by the ability to share the right information with the right partners at the right time. While this interconnectedness and situational awareness is crucial to success, it also opens the possibility for misuse of the same capabilities by sophisticated adversaries to spread attacks and exfiltrate or corrupt critical sensitive information. We have been investigating means to analyze behaviors of actors and assess trustworthiness of information to support real-time cyber security decision making through a concept called Behavior-Based Access Control (BBAC). The work described in this paper focuses on the statistical machine learning techniques used in BBAC to make predictions about the intent of actors establishing TCP connections and issuing HTTP requests. We discuss pragmatic challenges and solutions we encountered in implementing and evaluating BBAC, discussing (a) the general concepts underlying BBAC, (b) challenges we have encountered in identifying suitable datasets, (c) mitigation strategies to cope with shortcomings in available data, (d) the combination of clustering and support vector machines for performing classification at scale, and (e) results from a number of scientific experiments. We also include expert commentary from Air Force stakeholders and describe current plans for transitioning BBAC capabilities into the Department of Defense together with lessons learned for the machine learning community.**

I. Problem of Interest

In current enterprise environments, information is becoming more readily accessible across a wide range of interconnected systems. However, trustworthiness of actors is not explicitly measured as part of the quality of information, allowing actors to operate unaware of how the latest security events may have impacted the trustworthiness of the information and the peers involved. This leads to situations where information producers give documents to untrustworthy consumers and consumers use information from non-reputable documents or producers.

This work was sponsored by the Air Force Research Laboratory (AFRL). Distribution A. Approved for public release; distribution unlimited (Case Number 88ABW-2012-6304).

While cyber security monitoring systems have significantly evolved over the last decade, these systems still face a number of limitations. First, currently deployed monitoring solutions tend to be signature-based and narrowly focused on specific parts of the overall systems. Examples include the Snort [1] network intrusion detection system or the Host Based Intrusion Detection System (HBSS) [2]. This leaves more sophisticated attacks unhandled, such as 0-day attacks for which signatures are unknown, or insider attacks that require correlation across systems and layers. Second, current access control is based on static policies that tie cryptographic credentials to attributes that are used by access control rules. Dynamic events, such as subversion of credentials (e.g., theft of a Common Access Card [3]) or changes in actor behaviors (e.g., insiders performing illegitimate actions within their privilege realm), are not addressed at all, leaving systems vulnerable for a considerable period. Finally, vast amounts of audit data are collected within enterprise environments in the form of server logs which could potentially play a role in access decisions, but this data is typically only used for offline forensics, leading to a situation where "later is too late."

Analysis of observable behaviors for the purpose of establishing trust models has been a rich research area in cyber security. Chandola et al. present a good survey covering anomaly detection work performed in the past decade [4]. A rich set of literature also exists on spotting behavior differences for the purpose of insider detection [5]. At the network level, a concept called Behavior-Based Network Access Control (BBNAC) performs clustering to identify network behavior profiles [6]. Machines are admitted to the network only if their profiles are deemed normal by their closest cluster of behavior. BBNAC faces many of the same challenges that Behavior-Based Access Control (BBAC) does, but BBAC faces additional challenges associated with performing analysis at multiple layers, including the network layer, application layer, and document

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2013		**00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Using Machine Learning for Behavior-Based Access Control: Scalable Anomaly Detection on TCP Connections and HTTP Requests	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **RaytheonBBN Technologies,10 Moulton Street,Cambridge,MA,02138**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
32rd Military Communications Conference (MILCOM 2013), San Diego, CA, November 18 -20, 2013.

14. ABSTRACT
Today?s business processes are more connected than ever before, driven by the ability to share the right information with the right partners at the right time. While this interconnectedness and situational awareness is crucial to success, it also opens the possibility for misuse of the same capabilities by sophisticated adversaries to spread attacks and exfiltrate or corrupt critical sensitive information. We have been investigating means to analyze behaviors of actors and assess trustworthiness of information to support real-time cyber security decision making through a concept called Behavior-Based Access Control (BBAC). The work described in this paper focuses on the statistical machine learning techniques used in BBAC to make predictions about the intent of actors establishing TCP connections and issuing HTTP requests. We discuss pragmatic challenges and solutions we encountered in implementing and evaluating BBAC, discussing (a) the general concepts underlying BBAC, (b) challenges we have encountered in identifying suitable datasets, (c) mitigation strategies to cope with shortcomings in available data, (d) the combination of clustering and support vector machines for performing classification at scale, and (e) results from a number of scientific experiments. We also include expert commentary from Air Force stakeholders and describe current plans for transitioning BBAC capabilities into the Department of Defense together with lessons learned for the machine learning community.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

layer. Multiple efforts have looked at analysis of application-level behaviors. Examples include the use of machine learning techniques by Ma et al. to predict whether or not HTTP requests are benign [7]. Multiple efforts have investigated the use of behaviors for authentication. Shi et al. use the behavior of cell phone usage to determine whether a cell phone has been lost and thereby changed owner [8]. Similar to BBAC, this work uses machine learning to learn a normal model of usage. However, no direct integration with access control functionality is provided. A recent DARPA (Defense Advanced Research Projects Agency) program called Active Authentication is also trying to significantly advance the state of the art in authentication frameworks for computer users by employing Artificial Intelligence techniques to determine the identity of the user based on observed behavior at multiple layers and time horizons.

The concepts and technologies developed as part of BBAC strive to overcome current limitations by means of three concepts that together enable accurate calculation of trustworthiness by virtue of reasoning over observables related to actors. First, BBAC combines explicit rule-based signatures of behaviors with statistical learning methods to derive effective and accurate classifiers that work well in dynamic environments. Rule-based systems excel at capturing subject matter expert knowledge and providing justification about trust value changes to operators. Statistical learning algorithms can successfully overcome problems associated with combining trust values assigned by different rules by learning the combination functions and adjusting them over time as environments change. Second, BBAC integrates trust assessments with existing access control schemes in a way that is guaranteed not to weaken existing policies. This not only helps with certification and accreditation found in government cyber environments, but also enables us to manage inaccuracies and uncertainties inherent in our trust assessment algorithms in a way that enhances the overall accuracy of access decisions. Third, BBAC is based on a multi-stage processing pipeline that spans multiple layers and separates heavy-duty computations, e.g., classifier training, from online event processing and inline interactions during access control checks. To scale up to enterprise regimes, BBAC combines clustering analysis with statistical classification in a way that maintains an adjustable number of classifiers, one per cluster.

The resulting impact of BBAC is to fix a major shortcoming of current cyber security mechanisms. Once authenticated, actors currently can operate with impunity within their authorization realms and, beyond audit trails, there is no systemic way to dynamically detect suspicious behavior and modulate access rights. BBAC diminishes the risk of misplaced trust, increases mission reliability and assurance, and deters abuse of authorized privileges. Similar in nature to credit card companies that monitor transactions in real-time and can preemptively block accounts, BBAC analyzes observable behaviors on a number of different layers in real-time to check for intricate trends that would otherwise go unnoticed. This enables BBAC to stop penetration of cyber attacks and exfiltration of sensitive data, e.g., stolen credit card or medical records, as they occur.

II. Data Preparation

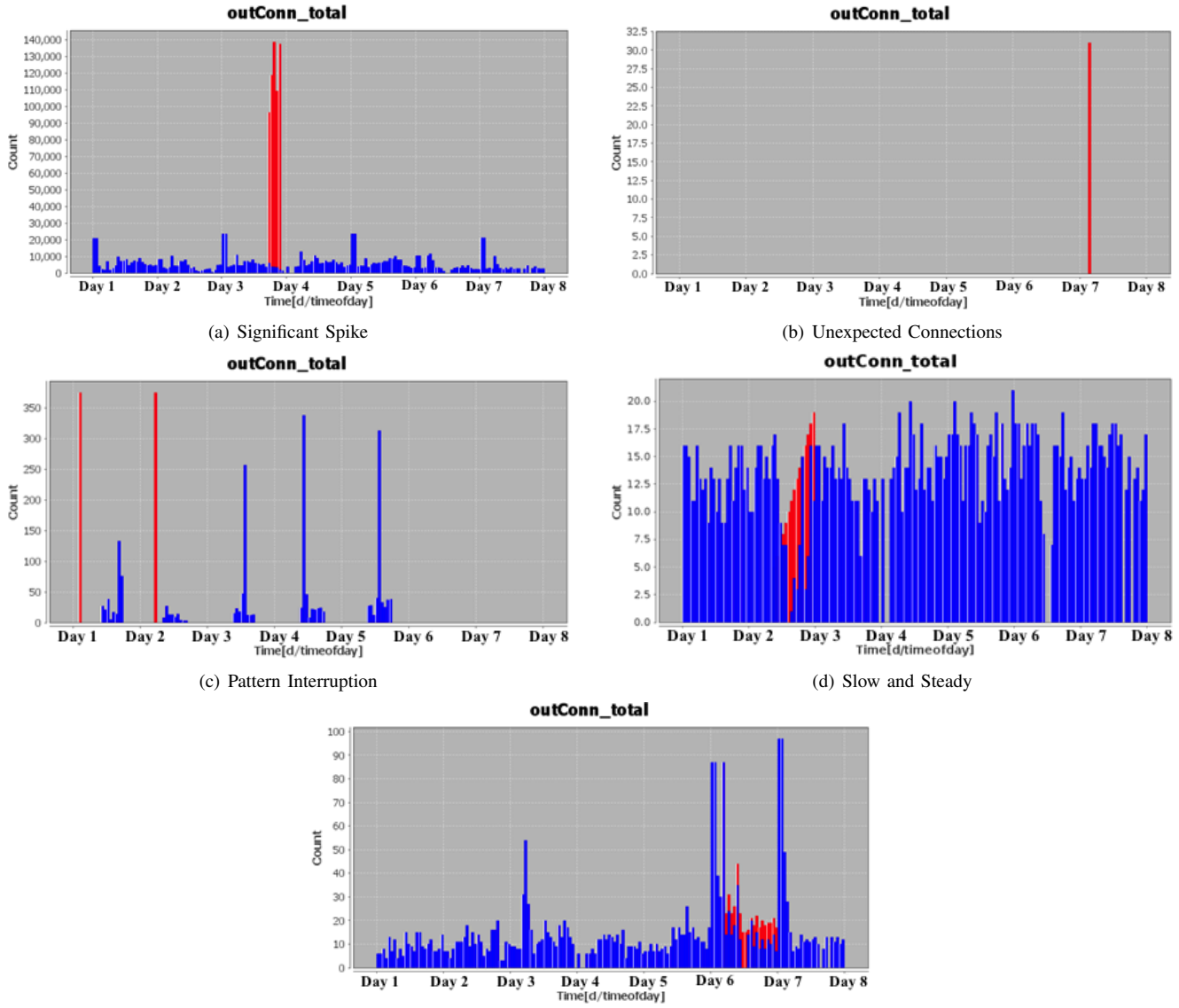
To ensure that BBAC is grounded in reality and can be used by cyber security operators to automate some of their tasks, we conducted a study looking at the types of data that are commonly available and used in enterprise environments today. We identified the following types of raw data: *Network flows*, containing IP addresses and ports of sources and destinations, timestamps, duration, and size, and *HTTP requests*, including URLs and HTTP headers.

This allows us to leverage logs from existing intrusion detection systems and services rather than introducing new mechanisms for collecting observables, which would face significant deployment cost and could add additional vulnerabilities to systems. While we were able to get access to gigabytes of data from these two sources, we encountered a number of problems.

Reluctance to share data before the value of analysis is clear. It seems as if the groups with the biggest need for advanced threat detection techniques are the ones who are least willing to offer data sets. There are multiple reasons for this. First, logs and audit records relevant for security analysis are generally very sensitive, as events frequently contain identifying information that can lead to breaches of privacy and complete sanitization of that information can be quite difficult. In addition, data providers might damage their reputation by giving out aggregate information showing that they are vulnerable to cyber threats. In essence, we face a chicken and egg problem with data providers only willing to give out data if the use of that data can be assessed and validated upon distribution time. This becomes a non-starter for R&D efforts that are looking to work and experiment with the data to evaluate new ideas.

Granularity mismatch. The behavior based analysis techniques we are investigating work best with data that has a rich context and feature space. What is needed is a large amount of granular data to do statistical inference. When looking at existing repositories, e.g., PREDICT [9], and deployed Host Based Intrusion Detection Systems (HIDSs), we encountered two different problems. First, in the case of PREDICT, we found examples of labeled attack instances, but they were very narrow in scope (packet captures), leading to few features, and only representative of a small number of specific attacks. Second, in the case of HIDSs, we found that we only had ready access to data preprocessed by correlator nodes. Getting access to more granular information, e.g., involving access patterns of processes on end-systems, would require installing software on end-systems or even recompiling applications (to map memory regions etc.), both of which raise practical concerns.

Lack of ground truth. To evaluate behaviors at the TCP level, we obtained a large dataset of BRO [10] network traces from the BBN network. One immediate problem is that very little can be linked to actual confirmed attacks, leading to the



(a) Significant Spike

(b) Unexpected Connections

(c) Pattern Interruption

(d) Slow and Steady

(e) Hide within Noise

Fig. 1. Simulated Attack Behaviors visible at the TCP layer. Blue lines indicate normal behavior and red lines indicate simulated attack behavior.

situation in which a large part of the traffic needs to get labeled as unsuspicious.

Size of available data sets. Real-world cyber security data sets tend to be rather large, producing hundreds of gigabytes of data per day even for smaller companies with less than 1000 actors.

Independence of data sets. Since BBAC performs analysis at multiple different system layers, we not only need access to data from sensors at these layers but the data in each layer needs to be linked to the other layers to represent a consistent picture of observables.

We developed a number of approaches to dealing with the various data problems. There is no proven claim about coverage or even success associated with the strategies at this point other than getting us over our immediate data hurdles.

Substitute simulation for ground truth. To address lack of ground truth concerns, we simulated a number of different attacks variants and observables based on how we would exfiltrate data or spread attacks at the TCP level if we were adversaries ourselves. This led to the development of the following randomized attacks (see Figure 1 for a visualization): **Category 1: Significant spike.** Significant consistent increase in outbound connections. **Category 2: Unexpected connections.** These attacks show outbound activity where there never was any, e.g., a server that has never made outbound requests suddenly making outbound requests. **Category 3: Pattern Interruption.** Many hosts follow a regular pattern (e.g., servers fetching updates at regular intervals). The attacks cause interruptions in those patterns. **Category 4: Slow and steady.** Slight increase over normal values, should still be detectable, though with lower accuracy. **Category 5: Hide within the noise.** These attacks form a control case, as BBAC

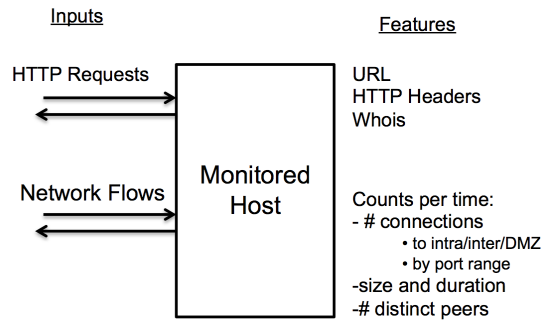


Fig. 2. TCP and HTTP features used to train our classifiers.

should not be able to detect them.

While the multi category attack simulations help with attack realism, concerns remain that the defenses we put in place are limited by our understanding of the attacks. Improving ground truth would be ideal, e.g., by tying the simulated attacks more directly to actual cyber events, such as the Stratfor Hack [11]. Another part of dealing with this problem is focusing on data that has ground truth. One such example is HTTP requests taken from black lists of known bad URLs.

Focus on realistically observable information. To address granularity issues, we focused analysis on data that is easily observable without new software or modifying end systems. We use the packet sniffer logs both for extracting features about TCP connections and HTTP requests.

Combine clustering with cloud computing to scale up. By first clustering, then training the SVM, cloud compute resources will enable scaling as the classifier for each cluster can be trained separately.

Splicing in attacks into normal traffic. To address the problem of independence between data sets, we developed an approach for injecting malicious URLs into request streams of benign hosts.

Red Teaming the feature space. We used Collaborative Red Teaming as an efficient means to get independent validation of claims about cyber defense in the past [12], [13]. To address concerns associated with attacks development, we devised simulation environments that enable fault injection and modeling of attacks at the feature level, thereby reducing the cost of attack development and allowing coverage over a wider range of attacks in feature space.

As a result of these mitigation strategies, we converted the raw observables into feature sets in a way that is not only reproducible but also can be directly hooked into deployed systems. The resulting features are summarized in Figure 2.

The reason for organizing network flow statistics (displayed at the bottom of Figure 2) by time of day and day of week is the understanding that behavior patterns differ over time and hence need to be represented in BBAC. Furthermore, aggregating the statistics into hourly bins seemed to provide a good trade off between spotting meaningful differences without getting lost in details. Our future work includes varying the size of the bins. For example, summarizing data into 15 minute bins and then aggregating the bins to summarize data over an hour or several hours. Binning allows us to look for patterns more easily but may also delay detection as data is binned. Differentiating between incoming and outgoing flows allows us to detect changes in patterns where servers (mostly incoming) turn into desktops (mostly outgoing). Changes in the location of peers (whether on the intranet, DMZ, or extranet), distribution of unique peers, size, and duration of flows might signal attacks, e.g., a machine that usually mostly communicates with internal machines all of a sudden reaching to many external machines and sending a large amount of traffic outbound.

For the HTTP data we collect similar features to Ma et al. [7] as shown in Figure 2. We have the following HTTP features: URL (also broken into parts: protocol, host, server, sub-domain, domain, path, query, reference), number of slashes in the URL, and number of queries in the URL. We also have access to the source id (an id that anonymizes the source IP address), source port, destination ip and port, date, user agent, method (e.g., GET, POST), referrer, mime type, MD5, status code and message. Additionally we have a feature for whether the host is an IP address or a string. We created a local WHOIS database [14] to pull in information about the age of a domain (features include: whether our local cache has the WHOIS information, whether the creation and expiration date is available and how long since/until the record was created/expires in days and months). We added WHOIS features about the domain age as many attacks use newly registered domains, e.g., for command and control servers.

To simulate HTTP attacks we used malicious URL feeds from http://www.phishtank.com, http://www.squidguard.org/ and http://dsi.ut-capitole.fr/blacklists/. Our attack data consists of normal HTTP requests with the URL replaced with a known malicious URL. We make the assumption that all URLs in our dataset from BBN are normal traffic based on conversations with the BBN network administrators who did not observe any attacks during this period. One of our next steps is to simulate additional attack vectors or signatures, such as information in the user agent header field.

III. Machine Learning Techniques

The goal of this work is to analyze the outbound HTTP requests and TCP connections that machines make to determine whether a machine has been compromised or an attack is executing. The theory behind this is that changes in the outbound traffic patterns could be indicative of insider data exfiltration, bots calling home, or an attack spreading. The features and methods we are using are primarily not novel (KMeans clustering and SVM SMO classifiers available in the Weka API [15]). The novelty stems from the exploration of how these classifiers perform using features from a deployed intrusion detection system on real enterprise data, the use of domain experts to synthesize attack data, and the combination of a clustering pre-processing step to both improve scalability and reduce false positives.

A. Clustering

We used the KMeans algorithm with KMeans++ initialization using the Weka API to cluster the host TCP data. The goal of the clustering is not to blindly create clusters – rather we want to create clusters based on aspects of a computer's behavior. Servers should be in different clusters than desktop machines. An administrative assistant is probably accessing different websites with different frequencies than a developer who is using cloud compute resources. By placing these machines into different clusters, we can learn what "normal" is for each one. Our intuition is that these clusters will map to different categories of machines (e.g., severs or desktops, web servers or web crawlers) allowing for improved classification rates and in particular fewer false positives. Multiple classifiers should be able to pick up behavior changes that a single classifier could not. For example, an attack vector might communicate using Twitter messages [16] – an infected server that only pulls in a nightly update would be very suspicious if it started using Twitter, however using Twitter is much less suspicious for a desktop machine.

We explored cluster sizes from 2 to 60 and performed clustering on what we call single instance data for each host. The single instance represents means and variances of the incoming and outgoing connections for each host and includes the total number of connections during work and non-work hours, and the average and standard deviation for both connection duration and size. We found that the cluster sizes generated by this approach exhibited a power law distribution. This distribution implies that there are several larger groups of machines with similar behavior and then many smaller groups with similar behavior. Unfortunately clustering machines in this way does not tell us how similar the groups are to each other. Our future work includes clustering using a combination of KMeans and decision trees to be able to describe the characteristics that a cluster of machines shares. To prevent overtraining, we want clusters that have a reasonably large number of machines, as opposed to clusters that only contain single machines. Ultimately, we clustered into 60 clusters, using the top 10 or 20 largest clusters along with a final cluster with all remaining hosts (for a total of 11 or 21 clusters total).

B. Supervised Learning

For the supervised learning approaches, we used support vector machines (SVMs) [17], using the SVM SMO [18] classifier via the Weka API. We selected it based on performance, efficiency, and availability of an easy-to-use, robust implementation.

The goal of SVMs is to find a hyperplane that separates a set of positive examples from a set of negative examples with maximum margin, using the dot product between two vectors. In many cases, using a non-linear classifier can outperform a linear approach. When this is so, a kernel function mapping the features to a higher dimensional space is useful. For most experiments, we used an RBF kernel with a gamma values between 0.01 and 100 (representing the width of the Gaussian). Varying gamma was one way to reach different points on the ROC curve. Training a support vector machine (finding the hyperplane with the maximum margin) requires solving or approximating a large quadratic programming (QP) problem. The SMO method decreases this training time by breaking down the QP problem into smaller problems containing only two Lagrange multipliers that can be solved analytically.

Some difficulties we experienced with this approach included mixed accuracy results based on the construction of the training set, the vast array of parameters available for tuning, and the fact that training times grow non-linearly with the number of observed hosts.

IV. Empirical Results

For both the TCP and HTTP data we replaced the BBN IP addresses with an integer identification number (we will refer to this feature as the host). This was done to preserve the anonymity of the BBN users. As we plan to combine our handling of the TCP and HTTP data, the identification numbers are the same in both data sets for the same IP address.

A. TCP data

Our generated attacks for TCP traffic create a spike in traffic with randomized start times, peak spike levels, and durations, for all hosts based on each host's original data (so far we have only tested with category 1 attacks). The values for these parameters were chosen from an uniform random distribution within a bounded range of values. Our classifier setup is described in Section III-B; we evaluated classifier accuracy using and RBF kernel, c (half log increments from 0.01 to 100), gamma (half log increments from 0.01 to 100), and a Weka spread subsample filter with values between 1 and 2 to compensate for more normal data than suspicious data.

There are two ways to train on this TCP data: a) *split by host id* by putting all the data from some hosts into the training set and all of the data from the other hosts into the test set, or b) *split by instance* by separating some data from each host into the training set and some into the test set. These two separations will train classifiers for two situations: a) the classifier is being trained to detect attacks for *new hosts* it has never seen data from before, or b) the classifier is being trained to detect attacks for hosts when it has seen both normal and attack data *for that host* before.

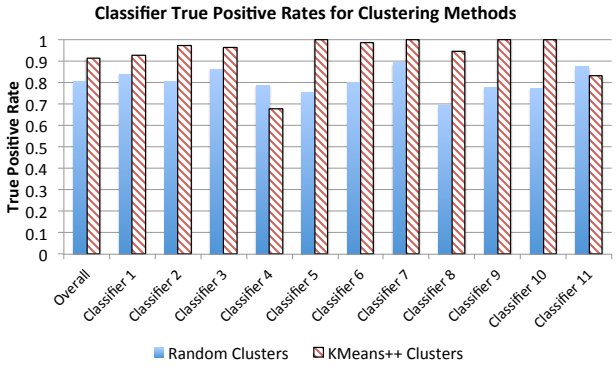
We measured accuracy by determining the True Positive (TP) rate, which measures the percentage of instances for which an attack instance got correctly classified as suspicious behavior, and the False Positive (FP) rate, measuring the percentage of instances for which a normal instance got incorrectly classified as suspicious behavior.

Not surprisingly when we tried a), our results were not good – although with some classifier settings (e.g., RBF kernel, $C = 0.5$, $gamma = 1$, $spreadsubsample = 1.2$) we could detect a few attacks accurately (25% TP, 0% FP). In essence this classifier is just learning a high threshold for traffic that will always be an attack – this technique will not work for more complex attacks (e.g., category 5: Hide within the noise) or for different spike sizes that are not as easily detected.

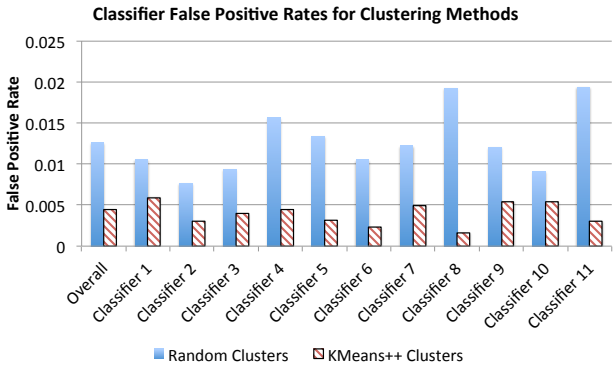
Using splitting technique b), we were able to achieve significantly higher TP rates (92%) but also higher FP rates (20%) with parameter settings $C = 10$, $gamma = 5$, and $spreadsubsample = 1.2$. We could detect many attacks, but also had a high incidence of false alarms. Alternatively we could achieve lower rates – TP of 34%, FP of 0% with $C = 5$, $gamma = 0.01$, $spreadsubsample = 2$.

Neither of these approaches is ideal as a deployed system must be able to detect both attacks in new hosts a) and deviations from past observed behavior b). Clustering and more narrowly targeted classifiers for each cluster may be able to accomplish both objectives: detect more attacks with lower false positive rates, as well as handle previously unseen hosts. Overall our goal is to be able to train a set of classifiers using different parameters and then intelligently select the best classifier for the circumstances the classifier will be operating in (e.g., high TP rate might be more important than a low FP rate if stopping all attacks is critical, while the opposite might be true if unrestricted access is critical).

As described in Section III-A we formed 11 or 21 clusters in this initial experiment. We then trained a classifier for each cluster using a subset of our data for each host in the cluster. This data included all features broken out by hour-of-day and day-of-week. In order to test that the KMeans++ clustering was effective, we compared the results with forming random clusters and analyzed the results using the Wilcoxon signed-rank test. The TP rates for the KMeans++ clustering were greater than the random clustering ($Z = 11.004, p < 0.001$). However, the FP rates were not significantly different ($Z = 0.369$). Although further analysis is needed, an initial examination shows that in cases where KMeans++ significantly improves the TP rate, the FP rate is also improved. Figure 3 shows the best 11-cluster results for both clustering methods. Further improvements to the TP and FP rates may be possible by adjusting the SVM parameters for each cluster instead of using the same parameters for all clusters.



(a) True Positive Cluster Rates



(b) False Positive Cluster Rates

Fig. 3. True positive (left) and false positive (right) rates for the best 11 cluster results for both random (solid blue) and KMeans++ (red stripes) clustering.

B. HTTP

We want to analyze our collected HTTP data to accurately identify malicious URLs. We envision the tool that results from this work operating in a proxy between the user and the Internet. This proxy could operate in different modes: either blocking suspicious requests, or allowing suspicious requests but notifying an administrator. Depending on operating conditions, it could be more critical to allow all traffic unless obviously bad (very low FP rate), or be conservative and block all traffic that was not clearly safe (very high TP rate). As such, trading off TP and FP rates in different situations is critical in this domain. In a typical week 17 million URLs are accessed in our medium-sized company. Even a 0.1% error rate that led to manual review would result in over 2000 URLs per day (1.7 per minute), a rate that is quickly infeasible.

The HTTP contains many string-based features, which are not compatible with SVMs. We tested different feature and filter combinations to determine how to best construct the HTTP classifier, including varying string conversation, feature subsets, and inclusion of WHOIS data. Our classifier was always a SMO classifier with a linear poly-kernel (exponent of 1) and a C of 1. We found that using an individual word vector for each string feature and using the WHOIS data provided the best result. So far we have been able to identify malicious URLs successfully with a rate of 99.6% with good URLs incorrectly classified as malicious at a rate of 1.0%. The classification time for a given URL is 45 ms with a standard deviation of 5 ms. Classification at this speed would be acceptable for an inline, real-time system. Retrieving WHOIS data in a real-time system needs to be done carefully – we have a local database of cached WHOIS data. We have found that using WHOIS information is important as many attacks utilize newly registered domains. Our method does not use as many features or the same dataset as [7], however the numbers are comparable (the best algorithm in Ma achieved about a 1.5% cumulative error rate).

C. Future work

Our future work includes several approaches to increasing accuracy. We plan to combine the TCP and HTTP data (we can correlate based on host). Additionally we will attempt to identify more ground truth information about each host – for example whether it is a server or a desktop machine. We believe this will help us make more accurate clustering deci-

sions and improve accuracy. We have obtained the countries that correspond to IP addresses, and our initial investigation indicates that this could be a valuable features, however our experiments do not yet include this feature. We plan to explore additional selection techniques for clusters, for example taking into account cluster similarity when selecting clusters.

V. Expert Commentary

The US Department of Defense (DoD) requires successful balancing between the ability to share information to those who need it with the responsibility to protect that information from those that should not have access. Information sharing includes access among many parties to successfully complete mission objectives. The inability to access critical information, as well as the as inappropriate release of critical information, could put mission success and the lives of our warfighters at risk. The dichotomy of information sharing drives the access control methodologies used in many operational systems today.

Automated machine learning systems, such as the one described in this paper, provide the ability to increase the granularity and precision of access control without unachievable degrees of manual intervention. These systems show great promise to improve both the flexibility and security of operational systems by adapting to factors such as behavior or need. While the static access control processes in use today are well known and widely implemented, many of their failings are well documented. For instance, the extreme difficulty in appropriately sharing information with unanticipated users, or reducing the damage that could be done by trusted users who choose to behave maliciously.

These same abilities are potentially useful in more areas, such as deciding what information is releasable to others who normally would not have access to the information processing capabilities the information resides within. Given that future missions are more likely to involve many members who may also be within joint or coalition environments, a more efficient capability to appropriately share information with these partners becomes even more critical.

VI. Potential Infusion

The DoD clearly recognizes cyber space as an operational domain in its own right, with cyber warfare missions executing to protect the national security goals. The DoD Strategy for Operating in Cyberspace [19] is the first DoD unified strategy for cyberspace and presents a new way forward for DoD's military, intelligence, and business operations. The strategy identifies five strategic initiatives that define a road map for the DoD to operate effectively and securely in cyberspace. Strategic initiative 2 specifically focuses on employing new defense operation concepts to protect DoD networks and systems based on the observation that intrusions might not always be prevented at network boundaries, and that advanced cyber capabilities need to be developed to support detection and mitigation of malicious activities performed on internal DoD networks by insiders.

BBAC contributes to the vision put forth by DoD seniors by providing increased real-time situational awareness in cyber space. Incidents such as Cablegate [20] show the impact of failing to catch data exfiltration attacks while they occur, leading to situations where checking audit records later is too late. In addition, advanced persistent threats posed by foreign nation states, e.g., as described in [21], demonstrate the level of sophistication outsiders can bring to bear to profit from cyber espionage. While BBAC is still in an early R&D development cycle and its prototype capabilities are only at a Technology Readiness Level of 2, it has shown the promise of significantly increasing work factors of such actors. The current BBAC prototype system was successfully shown to a number of DoD cyber security stake holders at the demonstration floor of the National Security Agency (NSA) Information Assurance Symposium (IAS) 2012, and spiked the interest of a number of potential transition partners. We currently see multiple avenues for increasing the Technology Readiness Level of BBAC to a point where it can be used operationally to increase defensive cyber capabilities within the DoD.

The first opportunity is to infuse BBAC into cyber security monitoring performed within the DoD. BBAC's use of machine learning is particularly promising in dealing with an increase in the sophistication of attacks that is expected to occur as the result of more consistent roll out and deployment of traditional security processes and protections, e.g., continuous monitoring and patching using the Secure Content Automation Protocol (SCAP) [22]. The combination of clustering and SVMs at multiple data layers promises to identify suspicious behaviors at scale, with high accuracy, and at an operation tempo that keeps up with rapidly executing attacks.

The second transition opportunity is centered around cross domain information sharing within the US or between the US and its allied partners. In that context, BBAC can be used as a service to perform behavior-based filtering on cross domain flows in a scalable and secure way.

VII. Lessons for the ML community

While security has been a domain where there has been considerable success for learning (from spam detection to Google safe browsing), it is crucial to consider how the learning algorithm will fit into the greater system. In a real world system, the results of the learning algorithm must be acted upon by programs or humans and the system must be retrained periodically with new data. Real world adversaries will attempt to evade or subvert the algorithm.

For a variety of reasons, it is difficult to get real attack data. However, merely training on noisy, normal data is insufficient. Therefore, it is important to work with domain experts that can help synthesize realistic attack data.

Real world traffic is heterogeneous, so using clustering as a pre-processing step shows promise for scalability and reducing false positives.

Ultimately, we believe that learning can have a real impact on cybersecurity in general and behavior-based access control in particular. However, we believe that most of the advances

that will result in real impact will be improving the training (data set creation), feature engineering, and situating the learning algorithm in the overall system, rather than the learning algorithm in particular. However, because of this fact, robust improvements to precision are likely to be most important to this space as high false positive rates can limit the utility of these systems.

The BBAC concept and algorithms described in this paper represent an important step in increasing the sophistication of cyber defenses to a point where we can discuss the possibilities of catching and mitigating sophisticated cyber attacks in real-time. The prototype implementations for analyzing network flows and web requests have served us well to demonstrate the concepts to DoD information assurance professionals and spurred interest from the community by showing concrete benefits of the system in terms that cyber operators can understand.

Going forward, we have a number of tasks aimed at improving the scalability and realism of BBAC. First, we have started to base the system design on a streaming cloud environment, using Storm [23] for performing distributed and fault-tolerant real-time computation. Storm is capable of processing millions of streaming tuples per second per node, enabling BBAC to scale up to DoD enterprise monitoring scenarios. Second, we will continue to engage transition partners for data sets, with a special focus of getting overlapping data sets. This will enable us to validate BBAC's capabilities of analyzing combinations of event streams from different sources at different system layers. Finally, we will define services and user interfaces for managing the training and classification activities to the point where they can be used by cyber operators. It is our belief that getting feedback from the hands-on use of the system by actual operators is critical to its adoption.

ACKNOWLEDGEMENTS

We acknowledge the support of various team members on the BBAC effort, including John Benner of Booz Allen Hamilton; Pavan Kantharaju, Andrew McDonald, Joe Muoio, and Jeffrey Segall, of Drexel University; and Jeffrey Cleveland of Raytheon BBN Technologies.

REFERENCES

[1] M. Roesch *et al.*, "Snort-lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration*. Seattle, Washington, 1999, pp. 229–238.

[2] DISA. (2012) The host based security system. [Online]. Available: http://www.disa.mil/Services/Information-Assurance/HBS/HBSS

[3] DoD. (2012) The common access card. [Online]. Available: http://www.cac.mil

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[5] M. Salem, S. Hershkop, and S. Stolfo, "A survey of insider attack detection research," *Insider Attack and Cyber Security*, pp. 69–90, 2008.

[6] V. Frias-Martinez, J. Sherrick, S. J. Stolfo, and A. D. Keromytis, "A network access control mechanism based on behavior profiles," in *Proceedings of the 2009 Annual Computer Security Applications Conference*, ser. ACSAC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 3–12. [Online]. Available: http://dx.doi.org/10.1109/ACSAC.2009.10

[7] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 30:1–30:24, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961202

[8] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *Proceedings of the 13th international conference on Information security*, ser. ISC'10. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 99–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=1949317.1949329

[9] RTI International. (2012) Protected repository for the defense of infrastructure against cyber threats (PREDICT). [Online]. Available: http://www.predict.org

[10] V. Paxson, "Bro: a system for detecting network intruders in real-time," in *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, ser. SSYM'98. Berkeley, CA, USA: USENIX Association, 1998, pp. 3–3. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267549.1267552

[11] G. Friedman. (2012, January) The hack on Stratfor. [Online]. Available: http://www.stratfor.com/weekly/hack-stratfor

[12] M. Atighetchi and J. Loyall, "Meaningful and flexible survivability assessments: Approach and practice," *CrossTalk - The Journal Of Defense Software Engineering, Special Issue System Assurance: Preparation and Promise*, March/April 2010.

[13] B. Wood and R. Duggan, "Red teaming of advanced information assurance concepts," in *DARPA DISCEX '00*, 2000, pp. 112 – 118.

[14] L. Daigle, "Whois protocol specification," 2004, rFC 3912.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[16] M. Farnum, "Bot herders using twitter for command and control," *Computerworld*, Aug. 2009.

[17] B.E. Boser, I.M. Guyon, and V.N. Vapnik, "A training algorithm for optimal margin classifiers," in *5th Annual ACM Workshop on COLT*, 1992.

[18] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, Tech. Rep., 1998.

[19] (2011, Jul.) Department of Defense strategy for operating in cyberspace. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a546341.pdf

[20] *US red-faced as 'cablegate' sparks global diplomatic crisis, courtesy of WikiLeaks*, Sydney Morning Herald, Nov. 2010.

[21] M. Riley and D. Lawrence. (2012, July) Hackers linked to China's army seen from EU to D.C. [Online]. Available: http://www.bloomberg.com/news/2012-07-26/china-hackers-hit-eu-point-man-and-d-c-with-byzantine-candor.html

[22] D. Waltermire, S. Quinn, K. Scarfone, and A. Halbardier, "The technical specification for the security content automation protocol (scap): Scap version 1.2," *NIST Special Publication*, vol. 800, p. 126, 2011.

[23] Storm. (2012, Oct.) Storm home page. [Online]. Available: http://storm-project.net/