Know Your Enemy, Know Yourself: Block-Level Network Behavior Profiling and Tracking

Esam Sharafuddin, Nan Jiang, Yu Jin, Zhi-Li Zhang
University of Minnesota
{shara,njiang,yjin,zhzhang}@cs.umn.edu

***Abstract*—Gaining a better knowledge of one's own network is crucial to effectively manage and secure today's large, diverse campus and enterprise networks. Because of the large number of IP addresses (or hosts) and the prevalent use of dynamic IP addresses, profiling and tracking individual hosts within such large networks may not be effective nor scalable. In this paper, we develop a novel methodology for capturing, characterizing, and tracking network activities at the *block-level* by carefully selecting a port feature vector and capturing the port activities of individual hosts within a block using a *block-wise (host) port activity matrix (BPAM)*. Applying the SVD low-rank approximation technique, we obtain a low-dimensional subspace representation which captures the significant and typical host activities of the block. Using these subspace representations, we cluster and classify blocks to provide high-level descriptive labels to assist network operators and security analysts to gain insight into the network activities. We also develop novel methods to track and quantify changes in blocks' behaviors over time, and demonstrate how these methods can be utilized to identify major changes and anomalies within the network.**

I. Introduction

Due to its scale and complexity, managing and securing today's large campus or enterprise networks is a challenging task. The scale and complexity comes not only from the number of heterogeneous hosts and devices on the network (e.g., various servers, desktop office client machines, laptops, lab machines, wireless access points, routers and so forth), but also from a wide range of diverse applications running on these machines. Traditionally, network security has largely focused on identifying and preventing attacks, e.g., through attack signature generation or anomaly detection. However, the scale, complexity and diversity of large campus and enterprise networks render such an approach *alone* less efficient, scalable and manageable. For instance, in the case of anomaly detection, what constitutes "anomalous" activities in one network or part of it may be considered "normal" in another network (or subnet). As an example, "unauthorized" peer-to-peer file sharing applications are not allowed on a departmental subnet of our campus network (unless they are for research purposes); on the other hand, such peer-to-peer activities are considered legitimate on student residential hall subnets. Hence, to more effectively manage and secure a large, diverse network, one must also build a good knowledge of one's own network, e.g., by understanding the range of applications, usage patterns and user behaviors in various parts of the network. Such knowledge will enable network operators and security analysts to better tailor their monitoring system and detection tools (e.g., firewall configurations), and focus their attention on specific vulnerabilities or areas of problem.

Along this new direction of *understanding oneself*, several research studies [1]–[3] have developed algorithms and tools for (primarily) *host-level* traffic classification and profiling. While these studies offer innovative methods for classifying traffic or host behaviors, the analysis at the granularity of individual hosts (or individual IP addresses) has two major drawbacks in practice. First, the prevalent usage of dynamic IP addresses makes tracking individual hosts an infeasible task in most networks [4], [5], since dynamic IP addresses are frequently reassigned to different hosts. Furthermore, the large number of IP addresses (e.g., our campus network has 3 class-B subnets, with 3×10^{16} potential hosts) make applying host-level traffic profiling to every host quite expensive.

To address these limitations, in this paper, we propose and develop a novel methodology for *block-level* network traffic behavior profiling. An IP address block constitutes a set of consecutive IP addresses, typically in size of 2^k, say, $k = 8$ (i.e., a /24 or class-C block), a unit used by a network administrator for IP address assignment to a subnet. More often than not, many hosts within the same block would be used for similar usage, e.g., a department block for office desktop and laptop machines, a block for lab machines, a block for student residential hall, a block with one or two wireless access points, and so on. As shown in [5], dynamic IP addresses are generally assigned in a block of consecutive IP addresses. Hence, by analyzing and profiling network activities at the *block-level*, we can circumvent the issues caused by dynamic IP addresses. Furthermore, by exploiting the similar user activities and usage patterns within a block, we can obtain a more compact block-level behavior profile which captures and summarizes the significant and typical behaviors of hosts within the block. Finally, the block-level analysis is far more scalable: in the case of our campus network, using /24 blocks, we only need to profile and track at most 768 $(= 3 \times 256)$ blocks as opposed to 3×10^6 IP addresses.

In this paper, we employ flow-level data (i.e., Netflow data) captured at the campus border router, and utilize the *port* information thereof to characterize and profile traffic behaviors and host activities at the block level. By considering well-known service ports, popular application ports and other dominant ports extracted from our flow data, we form a *port feature vector* consisting of 2000 source and 2000 destination

The work is supported in part by the National Science Foundation grants CNS-0626808, CNS-0626812, CNS-0905037 and the DTRA grant HDTRA1-09-1-0050.

ports. Using this port feature vector, a straightforward way to summarize the behavior of a block is to simply compute the aggregate port distribution of the block: namely, for each source or destination port in the port feature vector, the fraction of flows using the port that are generated by any IP address (or host[1]) within the block. However, while the aggregate port distribution captures the overall activities of hosts within the block, it fails to provide adequate information to capture, characterize, and distinguish significant and typical host behaviors within the block. For example, we would like a *block-level behavior profile* to enable us to meaningfully answer questions such as the following: i) Do all hosts in the block behave similarly, e.g., most of them are client machines that are used to primarily access the web? Thus, the overall port distribution would represent the "typical" behavior of the hosts in the block. ii) Does the block contain one or a few dominant hosts (e.g., web servers, or "heavy-hitter" client machines) that generate a majority of the flows? In other words, the overall port distribution of the block is skewed mostly by these dominant hosts, while obscuring the activities of other "typical" hosts within the block. iii) Or, does the block consist of several groups of hosts with distinct behaviors or activities, e.g., web/email servers, client machines with heavy web and P2P activities?

The ability to answer these and similar questions is important to characterize, summarize and distinguish the behaviors of various inside (campus) blocks within a network, and therefore help network operators and security analysts to understand and monitor the block-level activities, detect sudden changes and anomalies, and identify policy violations, security breaches and malicious attacks. For this purpose, we introduce the *block-wise (host) port activity matrix* (BPAM), which records the activities of each host within the block on these ports, i.e., the number of flows using each of the ports. Hence, the BPAM represents the key port activities of individual hosts within each block. By applying the Singular Value Decomposition (SVD) method (to an appropriately normalized and re-scaled version of BPAM), we obtain a compact, low-dimension *subspace representation* of the behaviors of each block. We show that as a low-rank approximation to the original BPAM, this subspace representation captures the *significant* and *typical* activities of individual hosts within the block, and can be used to answer the questions listed above.

Moreover, by introducing a subspace distance metric, we employ the subspace representations to cluster and classify the behaviors of various blocks of the network. The block-level behavior clustering allows us to assign *interpretive* labels to various blocks as to assist network operators and security analysts in understanding the overall block-level activities within the network. Furthermore, we demonstrate how to use the subspace representations to track changes in block-level behaviors over time, and develop two methods to quantify and classify such changes. We also show how these methods can be explored to identify major changes and anomalies. The efficacy of our proposed block-level network behavior profiling methodology has been extensively evaluated and validated using a month-long netflow data collected at our campus network.

[1]For simplicity, in this paper we use the term *host* to denote a specific IP address (although an IP address may be assigned to a router, a printer or some other devices).

Related Work. Several approaches have addressed traffic classification. Unlike [6]–[8] which rely on packet payload, our approach utilizes netflow data captured at a vantage point which is less expensive. While [1]–[3] classify traffic on the host-level, our approach; on the other hand, performs traffic classification at the block-level which is more scalable especially for large campus or enterprise networks with larger number of IP addresses, a good share of which are dynamic. Moreover, our scheme provides compact and summarized descriptive labels for the *significant* and *typical* activities a block. While [9] uses closely related methodology of PCA to describe structures of OD flows and utilizes the approach to detect specific types of attacks, our work compactly summarizes the block behaviors and its underlying activities.

More closely related work, the authors in [10] utilize host-port associations and apply probabilistic latent semantic analysis (pLSA) to extract *activity patterns* and provide a *global view* of the activity patterns within the network. In this work, we capture the *significant* and *dominant* activities within the block based on its subspace representation and provide meaningful labelling for each block .

The remainder of this paper is organized as follows. In section II, we describe how the port feature vector is selected. In section III, we introduce the BPAM and the SVD-based subspace method to extract and summarize significant and typical host behaviors within each block. In section IV, we develop a clustering method to classify block-level behaviors using the subspace representations, while in section V, we develop methods for tracking and quantifying block-level behavior changes, and show how they can be used to identify anomalies. The paper is concluded in section VI.

II. Block Behaviors and Port Feature Vector

Port numbers are the most widely used packet/flow level features for identifying network activities. Certain (IANA reserved or registered) ports are almost synonymous with the well-known services associated with these ports, e.g., web with TCP 80, email with TCP port 25, etc. Although these reserved ports may be misused by other applications (e.g., for penetrating firewalls), or the well-known services may also use other ports (e.g., TCP 8080 for web), for a majority of the hosts, the dominant activities observed on these ports represent the well-known services. Hence, such dominant service ports often reflect the typical server/client activities in each block. In this section, we utilize this observation to select a set of frequently used ports to form a *port feature vector* for characterizing block-level network activities. Before presenting this method, we first describe the datasets used in our study.

Dataset. Our study is based on a one-month data collected at the border router of our campus network. The data includes bidirectional Cisco NetFlow records corresponding to traffic between inside (campus) hosts (3 class B IP blocks with 2^{16} IP

addresses) and outside hosts. We only focus on the *outgoing* TCP, UDP and ICMP traffic which account for more than 99% of all the outgoing traffic, since incoming traffic usually contain a significant amount of "noise" (e.g., scanning), which may not even pass the border firewall. We note that we choose the block size to be 2^8 (class C IP block) throughout the paper, which is the most commonly used block size for our network administrators to assign IP addresses to different departments/subnets [5].

Port Feature Vector Selection Our port selection method is as follows. Let F_t be the set of flows observed during a time interval t, say, a day (this is the time interval used in this paper). We rank all the source (resp. destination) ports that appear in F_t in terms of both the number of flows containing the ports and the number of hosts with flows containing the ports. We pick the top ranked N source ports and N destination ports in such a manner that they cumulatively cover at least, say 95%, of all flows as well as of all "active" hosts (an IP address which generates at least one flow during the time interval t). Through experiments using our flow datasets, we decide on $N = 1999$, which yields 1999 top source ports as well as 1999 top destinations[2], and these ports cover nearly 98% of all the flows and close to 100% of all "active" hosts.

In addition, we group all remaining source/destination ports as if they were special "virtual" source/destination ports, referred to as "all other source/destination ports" (or *aoSrcPort*/*aoDstPort* in short). We define a 4000-dimension *port feature vector* $PFV = [port_1, \ldots, port_{4000}]$, where for $1 \leq j \leq 1999$, $port_j$ refers to one of the top 1999 source ports (ordered in the increasing number of the port numbers), and $port_{2000}$ refers to *aoSrcPort*; and for $2001 \leq j \leq 3999$, $port_j$ refers to one of the top 1999 destination ports (ordered in the increasing number of the port numbers), and $port_{4000}$ refers to *aoDstPort*.

These dominant ports indeed reveal the popular activities in each block. For example, in blocks belonging to certain department subnets which maintain their own web servers and email servers, service activities like web (80 and 443), ssh (22) and email (25 and 993) account for a high percentage of flows. In contrast, in residential halls blocks, client activities are more popular, with web (80 and 443) and messenger (AOL for 5190 and MSN for 1863) being the dominant ports.

III. Block-wise Host Port Activities and Subspace Representation

In this section, we introduce the *block-wise (host) port activity matrix* (BPAM) to represent the port activities of individual hosts within a block. Using the SVD method, we derive a low-rank approximation to the BPAM, and thus obtain a low-dimension *subspace representation* of the behaviors of each block. Applying this technique to our campus flow data, we demonstrate that these subspaces indeed capture the *significant* and *typical* activities of individual hosts within a block that can be used to meaningfully characterize and distinguish various block behaviors within a network.

[2]We refer to ICMP type numbers as the port numbers for ICMP traffic, e.g., ICMP0 represents ICMP type 0 echo reply traffic.

Definition of BPAM. Given a /24 block B, let i, $0 \leq i \leq m-1$ (where $m = 256$), denote a host (more precisely an IP address) within the block. For each host i, let f_i denote the total number of (outgoing) flows generated by host i during an observation time window t, say, a day. We use j to represent the jth port in the port feature vector that we define in the previous section. Then f_{ij} denotes the number of flows generated by host i using source port j ($1 \leq j \leq 2000$) or targeting destination port j ($2001 \leq j \leq 4000$), where $\sum_{j=1}^{2000} f_{ij} = \sum_{j=2001}^{4000} f_{ij} = f_i$. We define the BPAM for a block B as a $m \times n$ matrix $F_B = [f_{ij}]$ (where $m = 256$ and $n = 4000$). Fig.1 visually depicts F_B for a block with predominantly client machines (a) and a block with many web and email servers (b), where each row in the figure corresponds to a host within the block, and a dot at the jth position of the ith row is plotted if and only if $f_{ij}/f_B \geq 0.005$ and $f_i \geq 100$ flows, where $f_B = \sum_{i=1}^{256} f_i$ is the total number of flows generated by all hosts in the block. The left half (1-2000) of the x-axis represents the source ports, where x=2000 represents *all other* source ports (*aoSrcport*), and the right half (2001-4000) represents the destination ports, where x=4000 *all other* destination ports (*aoDstPort*). These figures visually illustrate that these blocks indeed exhibit distinct behaviors characteristics.

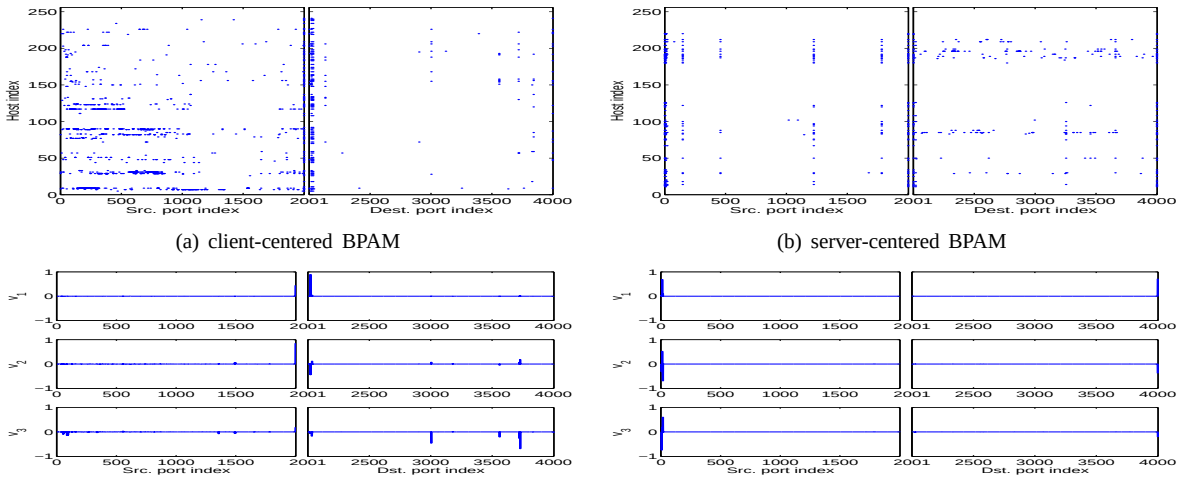
To reduce the impact of "heavy-hitter hosts" and "inactive" hosts within a block B, we introduce an appropriately *normalized* and *re-scaled* version of F_B defined as $A_B := [s_i p_{ij}]_{m \times n} = [s_i f_{ij}/f_i]_{m \times n}$. $p_i = f_i/f_B$ is the fraction of flows generated by host i and $ent_B := -\sum_{i=1}^{m} p_i \log p_i$, $0 \leq ent_B \leq \log m$: is the *(flow activity) entropy* and the closer ent_B is to the upper bound $\log m$, the more uniformly distributed are the flows among the hosts; whereas the closer ent_B is to 0, the more skewed is the flow distribution among the hosts. $s_i := ent_B/(-logp_i)$ if $p_i > 0$, and $s_i = 0$ otherwise[3]. We see that the smaller p_i is, the smaller s_i is. On the other hand, s_i only grows inverse logarithmically with f_B/f_i (approximately logarithmically with f_i), thus dampening the effect of extremely active hosts.

Extracting Typical Host Behaviors from BPAM. Given this definition of BPAM, we apply SVD [4] to A_B to extract "significant" and "typical" host behaviors of a block. We denote the rank-K approximation of the BPAM matrix A_B as: $A \approx U_K \Sigma_K V_K^T$, where $U_K = [u_1, \cdots, u_K]$, $\Sigma_K = diag[\sigma_1, \cdots, \sigma_K]$, and $V_K = [v_1, \cdots, v_K]$.

Intuitively, the right singular vectors, v_i's, provide an *orthogonal* representation of the port activities of the m hosts within a block. In a sense, each v_i can be viewed as a *virtual* host, where $|v_{ij}|$ (or v_{ij}^2) measures the magnitude of activities on port j by this virtual host (note that $\sum_{j=1}^{n} v_{ij}^2 = 1$; hence, $\{v_{ij}^2\}$ can be viewed as a probability distribution). Unlike the port activities of the m original (*real*) hosts, the m *virtual* hosts have *orthogonal* port behaviors (i.e., $v_i^T v_j = 0$ for $i \neq j$).

[3]Here we implicitly assume that each block has at least two active hosts, thus $p_i < 1$ for all i's. As $-\log p_i$ is the entropy of an individual host i, intuitively s_i measures the contribution of individual hosts' entropies to the (average) entropy of the block.

[4]We conducted the same experiments using robust PCA [11], [12], and the results are similar, possibly due to the normalization process which eliminates the effect of outliers.



(a) client-centered BPAM

(b) server-centered BPAM

(c) client-centered subspace representation

(d) server-centered subspace representation

Fig. 1. Block-wise host port activities for two example blocks and their subspace representations.

We refer to the *subspace* spanned by the first K dominant virtual hosts, $v_1, \ldots, v_K$, as a subspace representation of the (significant and typical) behaviors of a block.

Significant Block Behaviors and Interpretations. We employ the standard scree plot method to choose K. With $\epsilon = 0.05$, most blocks have at least 95% of the energy in the original BPAM captured by the first few singular values. For the two blocks in Fig. 1(a,b), we obtain $K = 3$ for both of them. In Figs. 1(c,d), we plot the "energy" of the ports, v_{ij}, in the top three singular vectors, v_1 (the top panel), v_2 (the middle panel), and v_3 (the bottom panel) for both blocks.

For the client-centered block in Fig. 1(c), we first note that most points with high energy (i.e., with large non-zero value v_{ij}^2) in v_1-v_3 are concentrated in the destination port range [2001-4000], except for one major point at $j = 2000$ (in v_1) corresponding to *aoSrcPort*. In addition to *aoSrcPort*, the other two largest points in v_1 correspond to destination port 80 ($j = 2020$) and port 443 ($j = 2033$). Hence, v_1 captures the web-related client activities of the hosts. Whereas, v_2 contains a number of large nonzero points corresponding to various destination service ports such as AOL Instant Messaging (IM) port 5190 ($j = 3728$), in addition to destination ports 80 and 443 (when the same point appears in both v_1 and v_2, it has opposite signs). v_3 also corresponds to other IM, such as Yahoo Messenger port 5050 ($j = 3721$), MSN Messenger port 1863 ($j = 3004$) in addition to remote desktop port 3389 ($j = 3561$). Therefore, the subspace spanned by the top three singular vectors captures the prevalent client behaviors of the hosts within the block, where web-related and IM activities are most significant and typical.

In contrast, for the server-centered block in Fig. 1(d), most points with high energy in both v_1 and v_2 are concentrated in the source port range [1-2000], except for the point at $j = 4000$ corresponding to *aoDstPort*. In v_1, the other major nonzero points correspond to major source service ports such as source ports 80 ($j = 11$) in v_1, whereas, v_2 contains a number of considerably large nonzero points in v_2 corresponding to various source ports such as email port 25 ($j = 8$), and ssh port 22 ($j = 7$). v_3 has similar ports (with opposite signs) to that of v_2. Hence, the subspace spanned by the top three singular vectors captures the prevalent server behaviors of the hosts within the block, where web, email, and ssh server activities dominate.

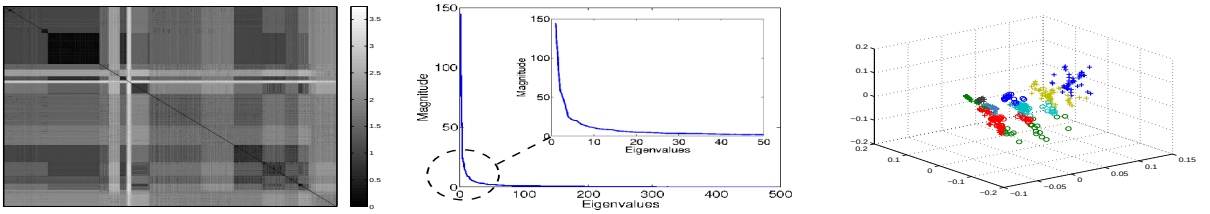
IV. Classifying Block Behaviors

The examples in the previous section show that blocks with hosts running different applications exhibit distinct behavioral characteristics, and the principal subspace provides a succinct way to extract, characterize and represent the significant and typical behaviors of a block. In this section, we compare and classify block-wise behaviors based on their subspace representations and group them based on their behaviors. Since $V^{(i)}$ is the resulting subspace representation of block B_i, we can utilize the subspace distance metric in [13] as follows: For any K_1-dimensional $V^{(1)}$ and K_2-dimensional $V^{(2)}$ subspaces in R^n, let $v_1^{(i)}, \ldots, v_{K_i}^{(i)}$ be an orthonormal basis of $V^{(i)}$, $i = 1, 2$. The *subspace distance* between $V^{(1)}$ and $V^{(2)}$ is $d(V^{(1)}, V^{(2)}) = \sqrt{\max(K_1, K_2) - \sum_{i=1}^{K_1} \sum_{j=1}^{K_2} (\langle v_i^{(1)}, v_j^{(2)} \rangle)^2}$. [13] shows that the subspace distance defined above is a Euclidean distance function, and *is independent of the choices of the orthonormal bases*, $v_1^{(i)}, \ldots, v_{K_i}^{(i)}$.

Using the above definition, we compute the pairwise subspace distance between any pair of the 492 blocks[5], and the results are shown in Fig. 2(a) which visually depicts groups of blocks that have shorter distances and have similar host behaviors. We then apply spectral clustering [14], [15], where the corresponding eigenvalues are shown in Fig. 2(b) and the clustering results shown in Fig. 2(c) for R^3 ($r = 3$).

The 10 clusters are represented by either "o" or "+" with different colors, where "+" stands for dense or seemingly dense clusters, and "o" stands for loose clusters. We have performed K-means clustering algorithms with different seeds, with $r = 3, \ldots, 6$, and $K = 5, \ldots, 25$. The overall observation remains the same: there are about 3 tightly clustered blocks,

[5]We only focus on 492 blocks which contains at least 10 active hosts with at least 10 observed flows originating from each host in a day.



(a) Visual representation of subspace distance matrix.

(b) Eigenvalue scree plot.

(c) 3-D embedding and K-means clustering.

Fig. 2. Clustering and Labeling of Blocks

ID	Label	Intra-dist.	Inter-dist.	#	Dominant Src. Ports	Dominant Dst. Ports
1	web client-centered	0.28	1.26	83	*aoSrcPort*	80
2	web server-centered	0.99	1.29	13	80, 25, 443	*aoDstPort*
3	non-web-dominated	1.01	1.42	28	Mail, p2p (no port 80)	*aoDstPort*
4	mixed web clients/web servers	1.1	1.59	51	80, *aoSrcPort*	80, *aoDstPort*
5	mixed web clients/servers	1.09	1.34	57	80, 25, 22	80, *aoDstPort*
6	diversified web clients	1.32	1.48	69	random ports	80, random ports
7	web and p2p clients	1.31	1.44	79	p2p	80 and p2p
8	mixed client behaviors	1.63	1.61	9	p2p and IM	80, p2p, mail and IM
9	mixed clients and servers	1.52	1.59	16	80, 25, IM and p2p	80,IM and p2p
10	outliers	1.64	2.13	6	special-service-ports	special-service-ports

TABLE I
SUMMARY OF CLUSTERING RESULTS, R=3, K=10

and a small group of "outliers" blocks, while other blocks belong to somewhat more loosely associated clusters. The number of clusters and membership of blocks hinge on the parameter K, the seeds, but less so on the dimension r. Table I summarizes sample clustering results. Each cluster is assigned a "high-level" *descriptive label*, based on the interpretation of common behaviors shared by most blocks in the cluster. The interpretation is derived by manually examining the ports with high *energy* (i.e., v_{ij}^2) in the top singular vectors of the blocks within a cluster. The blocks within the first three clusters are most tightly clustered, exhibiting more coherent behaviors. They correspond to predominantly client blocks, predominantly server blocks and blocks which belong to lab machines (e.g., CSE and IT labs) and the supercomputing center, where users of these machines do not routinely use them for web surfing.

The fourth cluster contains blocks comprised of client machines with predominantly web surfing activities together with at least one web server which is similar to blocks within the fifth cluster with the difference in that blocks within the fifth cluster include mail and ssh servers in addition to web servers. Behavior of blocks within the sixth cluster include clients that initiate web connections using random, yet frequently-used ports in our campus traffic. Blocks within the next three clusters are more loosely clustered and their behavior is characterized by far less dominant web client activities and no dominant web server activities. These clusters include blocks with various source and destination p2p ports, or IM or both. The last cluster contains blocks in which their behaviors are quite distinct due to being used for special purposes such as PlanetLab or machines used to conduct credit card transactions. [16] provides details of these clusters.

We conclude this section by emphasizing that the goal of our clustering of block behaviors is not to generate a precise classification, but to produce some high-level "descriptive" labels and provide a "big picture" view of the block-wise behaviors in a campus/enterprise network so as to assist network operators and security analysts to better monitor and manage the network. We have conducted the same study for data collected from an enterprise network and were able to articulate similar high-level "descriptive" labels, but these results are not presented due to space limitation.

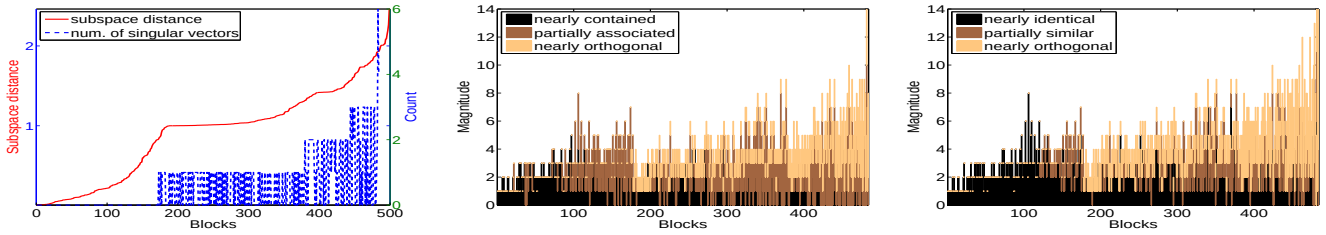
V. DETECTING CHANGE IN BLOCK BEHAVIOR

In this section, we show how to utilize the subspace representation to detect change in a block's behavior over time which may be indicative of potentially suspicious activities.

Methods for Tracking Behavior Changes over Time. Given a block B, let $V^{(t)}$ denote the subspace representation of its behaviors at the tth time interval (say, the tth day). We can use the subspace distance, $d(V^{(t)}, V^{(t+1)})$, to compare and track the change in the behavior of block B over time. Let $V^{(t)} = [v_1, \ldots, v_K]$ and $V^{(t+1)} = [w_1, \ldots, w_L]$. Clearly, if $K \neq L$, then $d(V, W) \geq 1$, a relatively large distance. Fig. 3(a) shows an example of the subspace distance of each of the 492 blocks in two consecutive days along with $|K-L|$. Generally, as $|K-L|$ increases, the subspace distance increases, however, for some blocks even though $K = L$, they still have relatively large distance which implies change in activities.

To have an insight on what has changed in the block's behaviors as evident by the distance, we develop two methods which provide more detailed information to quantify and track the behavioral dynamics of blocks over time. In the first method, we compute the distance of each singular vector w_i at time $t+1$ to the entire subspace V at time t and based on the distance, we label w_i as either *nearly contained* (distance close to 0), *partially associated* or *nearly orthogonal*. In the second method, we go one step further by comparing distance between individual w_i's to v_i's for the same block and recursively select (and remove) pairs in the order of increasing pairwise distances and label them in a similar fashion. (please refer to [16] for more details).

The results of applying these two methods are shown in Fig. 3(b) and Fig. 3(c), respectively, where the number of



(a) Correlation between subspace distance and change in number of singular vectors

(b) Method 1: Distance between a singular vector at t and subspace representation at $t+1$

(c) Method 2: Distance between pairs of singular vectors corresponding t and $t+1$

Fig. 3. Subspace distance between two consecutive days

w_i's or (w_i^*, v_i^*) pairs belong to each of the three categories is plotted in which the x-axis is fixed in the same order (i.e., increasing subspace distance) as in Fig. 3(a). As is clear from Fig. 3(b) and Fig. 3(c), for blocks with very small subspace distance all w_i's are *nearly contained* in V of the previous day (the dark shaded area under the curve in Fig. 3(b)). Furthermore, most "best-match" vector pairs (v_i^*, w_i^*)'s are *nearly identical* (the dark shaded area under the curve in Fig. 3(c)), with the remaining pairs at least *partially similar*. As the subspace increases, more w_j's become *partially associated* with V, suggesting that w_i's capture some new activities or changes in behaviors in the underlying hosts of these blocks and for cases where $K \neq L$, at least one w_i or at least one pair (v_k^*, w_k^*) is *nearly orthogonal*. Having more *nearly orthogonal* w_i or (v_k^*, w_k^*) suggests a more drastic change in the block's behavior.

We have performed similar analysis to compare, quantify and track the changes in block behaviors over time using two-week long data, and obtained qualitatively similar results that enabled us to identify specific activities that cause major changes. Due to space limitation, we do not present them here.

Detection of Suspicious Activities. We show how the methods we have developed above can be used to detect potential anomalies and attacks. We demonstrate this capability through *attack emulation*, where we inject certain types of attacks or other anomalous activities into a block with otherwise "normal" activities. We have performed this study using a range of anomalies including *outside scanning*, *back-door trojan activities* and *ddos attacks*. We notice that change in subspace representation caused by new activity in the block is rather minor compared to the more drastic change caused by malicious activities transforming the block's behaviors into completely different behaviors in which a source or a destination port suddenly becomes most significant and dominant. Full discussion of these emulated attacks can be found in [16].

Clearly, the resulting anomaly and attack detection uncovered by tracking the subspace distance of a block over time is only meant for *post-mortem* analysis or "after-the-fact" discovery of attacks or anomalies, not *real-time* detection. Thus, it is complementary to firewalls and other IDS (intrusion detection system) and IPS (intrusion prevention system) that are commonly deployed in large campus and enterprise networks. Our technique is particularly useful in uncovering compromised *inside hosts* that are (frequently, periodically or even occasionally) used to launch attacks or other illegitimate activities targeted at the outside Internet.

VI. Conclusions

In this paper, we developed a novel methodology for profiling and tracking network activities at the *block-level*. We applied methods to capture significant and typical host behavioral activities of the block to help us cluster and provide high-level descriptive labels of blocks. We also developed methods to track and quantify changes in block's behaviors over time, and demonstrated the use of these methods to identify major changes and anomalies. The proposed method was validated using a month-long flow data captured at the border router of a large university campus network.

References

[1] K. Xu, Z. Zhang and S. Bhattacharyya. Profiling Internet backbone traffic: behavior models and applications. In *Proc. of ACM SIGCOMM*, August 2005.

[2] T. Karagiannis, K. Papagiannaki and M. Faloutsos. BLINC: Multilevel traffic classification in the dark. In *Proc. of ACM SIGCOMM*, August 2005.

[3] K. Xu, Z. Zhang, and S. Bhattacharyya. Internet traffic behavior profiling for network security monitoring. In *IEEE/ACM Trans. Netw.*, 2008.

[4] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *Proc. of ACM SIGCOMM*, 2007.

[5] Y. Jin and E. Sharafuddin and Z. Zhang. Identifying dynamic IP address blocks serendipitously through background scanning Traffic. In *Proc. of ACM CoNext'07*, 2007.

[6] P. Haffner, S. Sen, O. Spatscheck, and D. Wang. ACAS: Automated construction of application signatures. In *Proc. of MineNet*, 2005.

[7] A. Moore and D. Zuev. Internet traffic classification using bayesian analysis techniques. In *Proc. of ACM SIGMETRICS*, 2005.

[8] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev.*, 37(1):5–16, 2007.

[9] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows, 2004.

[10] E. Sharafuddin, Y. Jin, N. Jiang, and Z. Zhang. Sifting through network data to cull activity patterns with heaps. In *Proc. of ICDCS*, 2010.

[11] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 35(1):109–120, 2007.

[12] B. Rubinstein et al. ANTIDOTE: Understanding and defending against poisoning of anomaly detectors. In *Proc. of ACM IMC*, 2009.

[13] L. Wang, X. Wang, and J. Feng. Subspace distance analysis with application to adaptive bayesian algorithm for face recognition. In *Pattern Recogn.*, volume 39, pages 456–464, 2006.

[14] U. von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[15] I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications*. Springer Series in Statistics, 2005.

[16] E. Sharafuddin and Y. Jin and N. Jiang and Z. Zhang. Block-level network behavior profiling and tracking. http://www-users.cs.umn.edu/∼shara/papers/TechRepNet09-1.pdf.