Profiling and Clustering Internet Hosts

Songjie Wei Jelena Mirkovic Ezra Kissel
Computer & Information Sciences Department
University of Delaware, Newark, DE 19716

Abstract— *Identifying groups of Internet hosts with a similar behavior is very useful for many applications of Internet security control, such as DDoS defense, worm and virus detection, detection of botnets, etc. There are two major difficulties for modeling host behavior correctly and efficiently: the huge number of overall entities, and the dynamics of each individual. In this paper, we present and formulate the Internet host profiling problem using the header data from public packet traces to select relevant features of frequently-seen hosts for profile creation, and using hierarchical clustering techniques on the profiles to build a dendrogram containing all the hosts. The well-known agglomerative algorithm is used to discover and combine similarly-behaved hosts into clusters, and domain-knowledge is used to analyze and evaluate clustering results. In this paper, we show the results of applying the proposed clustering approach to a data set from NLANR-PMA Internet traffic archive with more than 60,000 active hosts. On this dataset, our approach successfully identifies clusters with significant and interpretable features. We next use the created host profiles to detect anomalous behavior during the Slammer worm spread. The experimental results show that our profiling and clustering approach can successfully detect Slammer outbreak and identify majority of infected hosts.*

Keywords: Internet, host behavior, profiling, clustering.

1. Introduction

Today's Internet is plagued with a wide range of security threats such as fast worm spreads and distributed denial-of-service attacks. These threats are usually detected too late, after they cause a considerable damage to the normal operation. Even after successful detection, defense mechanisms are frequently challenged by the task of separating the legitimate from the attack traffic, since these two streams are highly similar.

Large-scale Internet security incidents introduce anomalies into the traffic patterns on the Internet backbones. Correct and rapid detection of these changes can help us detect Internet anomalies in time, so that effective measures can be carried out to prevent and fight potential cyber attacks. It is difficult to devise a permanent model of legitimate or anomalous host behavior, applicable to every Internet location, because of the heterogeneity of Internet hosts and the dynamics of their activities. On the other hand, each individual host and its users exhibit slowly-changing patterns of the Internet use over a relatively long period of time. We thus believe that single-host behavior profiles map out a promising direction for detecting Internet anomalies.

In this paper we investigate a problem of using public traffic traces to define host behavior profiles and categorize hosts by applying clustering techniques. The resulting clusters are used to detect anomalous host behaviors and flag such hosts as suspicious. In our future work we plan to assign some suspicious points to each host with an anomalous behavior, and use these points to shape an access or a traffic handling policy. Individual host profiles are likely to be more sensitive to anomalies than if we built a legitimate behavior profile for a generic host, and should aid early detection of stealthy threats such as slow-spreading worms or botnet recruitment.

The proposed profiling and host characterizations are applicable to any monitoring site, but they are likely to produce more useful results if applied at the backbone than at the edge, since many more hosts are observable at the backbone and their behaviors can be correlated and used to infer global behavior patterns. However, since public backbone traces only record short daily snapshots, we used public edge traces to demonstrate the feasibility of our approach. In our future work we plan to investigate how traffic sampling, present in public backbone traces, affects the precision of the host profiles and the clustering approach, and we show some preliminary results of this investigation in section 4-1.

The profiling and clustering of the Internet hosts are the first steps in our research on an Internet-wide host reputation system called Internet Credit Report (ICR). Just like the credit-reporting agencies, ICR would monitor Internet-wide activity and assign each host a reputation score based on its behavior. The knowledge provided by a host's reputation score about long-term good clients and recurring offenders would help improve Internet security and prioritize traffic during distributed denial-of-service attacks or worm spreads. The key insight behind ICR is that a given host tends to be well-administered or poorly-administered over a considerable time, and that hosts that have behaved maliciously in the past warrant a lower trust since they are likely to be compromised in the future. Research on host scanning patterns [2] has revealed that a few hosts are responsible for a large fraction of overall Internet scans and that large scanners persist over a considerably long time [2].

In Section 2, we present our approach to building host profiles. We describe the clustering algorithm for grouping hosts with similar behaviors in Section 3. In Section 4 we illustrate, through experiments, possible applications of host profiles for host categorization and anomaly detection. We survey related work in Section 5 and present conclusions and future work in Section 6.

2. Creating Host Profiles

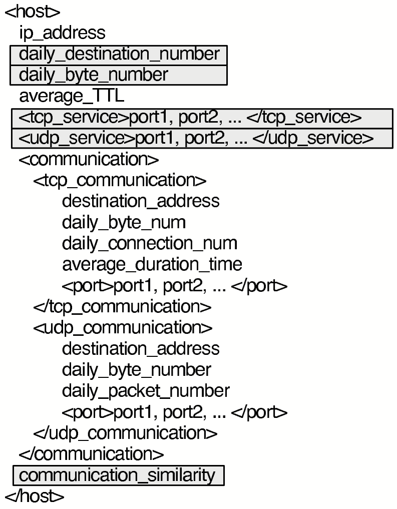
There are several challenges to be addressed for profiling Internet hosts at a large scale, especially using high-volume, diverse Internet traffic. The first challenge lies in the number of active hosts (identified by different IP addresses) observable in the backbone traffic traces, which can be several million. On the other hand, many observed hosts appear only sporadically, producing too scarce data for a useful profile. It is necessary to distinguish active hosts (such as an office desktop computer) from inactive ones (e.g., a Honeynet computer that receives a lot of traffic but does not initiate communication). Only the active host's traffic produces valuable behavior profiles, that can be further imporved using the inactive host's traffic. The second challenge lies in the dynamics of the host behaviors. Even given a single host, its behavior may change from time to time, for legitimate reasons, e.g., a user has discovered online gaming. This problem is more prominent when we observe Internet usage of many hosts, which exhibits burst behavior. In the rest of this section we describe our approach for creating host-behavior profiles, while carefully addressing the challenges of separation of active and inactive hosts, host-behavior dynamics, and the integration of traffic data collected at different times into host profiles.

2.1. Host Behavior Characterization

We use only packet header information, which is available in a sanitized form in public Internet traffic traces, to infer host characteristics. From packet headers, we obtain *direct* and *indirect* features for each host. Direct features are those that can be retrieved from a packet header without further computation, like the destination IP address and port number, the observed TTL value, etc. Indirect features include those computed using multiple packets in a host's communication, e.g. the average duration and traffic volume of a TCP connection. In our host feature computation, we make distinction between an active and a passive TCP communication. An active TCP communication of a given host consists of connections initiated by this host (by sending a TCP-SYN packet). A passive TCP communication consists of connections initiated by other hosts with a given host. Only active TCP communications are used for host characterization. For UDP traffic, each communication is listed as active for both the source and the destination hosts. Currently, we use one-day and two-day intervals for profile-building. With more detailed traces, shorter periods (e.g., one hour) could also be used.

The host features we extract for host behavior characterization are shown in the Figure 1, in an XML-like format:

- *ip_address:* the IPv4 address of the profiled host
- *daily_destination_number:* the number of distinct IP addresses contacted by this host.



```
<host>
 ip_address
 daily_destination_number
 daily_byte_number
 average_TTL
 <tcp_service>port1, port2, ... </tcp_service>
 <udp_service>port1, port2, ... </udp_service>
 <communication>
   <tcp_communication>
       destination_address
       daily_byte_num
       daily_connection_num
       average_duration_time
       <port>port1, port2, ... </port>
   </tcp_communication>
   <udp_communication>
       destination_address
       daily_byte_number
       daily_packet_number
       <port>port1, port2, ... </port>
   </udp_communication>
 </communication>
 communication_similarity
</host>
```

Fig. 1. Features used for host profiles

- *daily_byte_number:* the total byte traffic volume sent from this host, including both TCP and UDP traffic.
- *average_TTL:* the average of TTL (time-to-live) values observed in the trace of this host, reflecting its relative Internet location with regard to the traffic monitor. Since Internet routes do not change rapidly at a large scale, the observed TTL should not greatly diverge from this average.
- *tcp_service:* and *udp_service:* list open ports on a host that, together with a communication profile, facilitate recognition of a host's functionality, e.g., a DNS server, a Web server, etc.
- *communication:* detailed specification of typical communications initiated by the profiled host, including the destination IP and daily traffic volume in bytes, the average duration (TCP) and the average number of packets (UDP)
- *communication similarity:* diversity of all the communications recorded in the <communication> field. We first calculate Dice similarity [3] of any two communications as:

$$sim(c_i, c_j) = \frac{1}{k} \cdot \sum_{n=1}^{k} \frac{2 \cdot c_{in} \cdot c_{jn}}{c_{in}^2 + c_{jn}^2}, \qquad (1)$$

  where k is the number of features for each communication (5 for TCP and 4 for UDP), and c_{in} is the value of n-th feature for the communication c_i. The communication similarity is computed as the average of Dice similarity values of all the communication pairs for this host.

2.2. Data Preprocessing

To generate meaningful information for creating host profiles, we must extract selected features from the public traffic traces. All the trace files we currently use are in widely used libpcap/winpcap format. We utilized CAIDA's CoralReef [4] API and developed a set of programs to produce detailed traffic information from trace files. This information is then further processed and aggregated to produce host features for the host profiles.

The output of data preprocessing stage consists of TCP and UDP traffic statistics for each source/destination pair that include information described in Figure 1. For each source IP, a list of contacted application ports are listed along with the number of connection requests, traffic rate, average packet size (for UDP traffic), and duration (for TCP connection). Below we describe some difficulties and how we overcome them in this data preprocessing phase.

- *Identifying host services:* TCP and UDP services listed per source are identified by observing packets to a service port that receive a response within 15 seconds. If there is no response within that interval, the packet is considered a scan. We obtain a list of port number assignments for well-known services from [5].
- *Identifying TCP connection:* Any new TCP-SYN packet is considered to start a new connection if it receives SYN-ACK reply. If TCP traffic is seen between two hosts without having encountered an initial SYN packet, it is counted as a separate TCP connection. Upon seeing a TCP-FIN packet, the TCP connection is considered terminated within a user-defined time, with the default of 5 seconds. Upon seeing a TCP-RST, we consider the TCP connection terminated immediately.

During the data preprocessing step, we identify hosts that are *frequently* and *actively* appearing in the traces, and select only these hosts for profiling. *Frequently appearing* means that a host should be present in multiple traces collected at different times. Currently, we only build profiles for hosts actively appearing in traces of more than two continuous days. *Actively appearing* means that a host also actively initiates communications. We use this criterion to filter out those hosts that are silent but receive a lot of incoming scans. These two selection criteria drastically reduce the number of hosts for profiling, improving scalability, and result in more useful and efficient host profiles. In the edge traces we use in our experiments, about 83% of hosts appear only sporadically or are passive hosts that cannot be used for profiling. This is expected because edge network's hosts communicate with many and diverse destinations, that will appear as passive hosts in the trace.

2.3. Updating Profiles

Our underlying assumption that motivates creation of host profiles is that Internet users have some settled habits and routines when using network resources, which are reflected in stable communication patterns in a host profile. Still there are many small divergences from a routine user behavior that create considerable dynamics observed in the traces, and must be incorporated in the host profiles. For example, the majority of a specific host's daily communications can come from its Web browsing on the destination port 80. But a user (or multiple users) of this host may browse different Web sites each day, so the host's profile should be updated daily to reflect such dynamics. At other times, the host behavior changes at a large scale and may be a sign of anomalous events affecting this host. For example, the traffic volume of a worm-infected host can rise suddenly and sharply, with many new connections being initiated to numerous destinations on the same port number. Such sharp behavior changes should be flagged as suspicious and not be used for the profile update.

For the quantitative host features shown in Figure 1, the Exponential Weighted Moving Average (EWMA) is used to integrate observed values with the profile, with a weight value of 0.25 for newly-observed data. For the communications' records in the profile, it would be impossible to accumulate all the records for each host over a long period. We currently maintain only the latest N communications with N varying for different applications. Communications with the same age are included in the profile using their traffic volume as a secondary criteria, with the preference for large-volume communications. We examine each host's behavior in the new trace and compare it with its current cluster (see section 3-2) before the profile update. If the host's new behavior is identified as extremely anomalous (which is defined based on a criterion of dissimilarity between host behavior in the new trace and its belonging cluster), it will not be used for profile update.

3. Cluster Extraction from Profiles

Host profiles are used to group hosts with similar features into clusters, with a final goal of building the characteristic models of host communication. There are several reasons for creating groups of similar hosts instead of modeling each host separately. First, if the profiles are to be built online and used for creating host reputation at backbone monitors, it is infeasible to monitor each packet at the backbone in a real time and use it for profile update. If packets were sampled, this would lead to inaccurate profiles of individual hosts. On the other hand, even though there are billions of hosts on the Internet and even more human users, many of them show similar communication patterns and there is virtually no information loss if we group their profiles into a common category. By grouping hosts into categories, hosts in the same category can validate and complement each other's behaviors and profiles. Here we use a reasonable assumption, validated through our experiments, that although individual host's behaviors change over time, the profile of a legitimate host tends to fall into the same category for a moderately long time. The second reason for grouping hosts into categories is to build models of legitimate Internet communications. These models can aid detection of suspicious changes in the backbone traffic, which are usually a sign of an Internet-wide security problem (e.g., worm, DDoS attack). Large-scale incidents can thus be detected through macroscopic observations. A third advantage of grouping similar hosts is that it addresses the scalability of the profiling approach, and facilitates host profiling at the Internet scale. By controlling the clustering process to produce clusters with different resolution and precision, the number of desired host categories on various network and host populations can be controlled. The resolution and precision requirements can vary depending on the requirement for clustering performance (how fast and frequently the clustering process should run) and storage availability (how much space is available for storing features of numerous cluster categories).

Since we do not have an advance knowledge of the exact number of possible host categories and of the defining features of each category, the clustering techniques in data mining come as an appropriate tool for host grouping. In the following discussion, we present our host clustering procedure based on a hierarchical algorithm. We will use the term ”cluster” to refer to a host category.

3.1. Profile and Cluster Distance Measure

Unlike learning during a classification process, where there is some a priori knowledge concerning the importance of each feature and features are used serially, the clustering process requires use of all the features simultaneously and feature weights have to be assigned by the user. In our host clustering process, the choices of host features and their importance (expressed as feature weight) are based both on the availability of data and on our experiences in characterizing network traffic. A straightforward approach for clustering host profiles, containing features shown in Figure 1, is to digitize each feature and use all of them for calculating the similarity between hosts while building clusters. This approach makes sense for some features that are invariant across hosts with similar behavior (e.g., the daily number of destinations a host communicates with) but not for others (e.g., TTL value of a host depends on its distance from a monitor which collects the trace; two hosts with the same TTL value may have very different behaviors[1]).

We are currently using five host features for clustering, shown in the Figure 1 within shaded rectangles. The distance measure for clustering is based on Dice coefficient defined in equation (1). This same equation is used for our inter-cluster distance measure, but with different interpretation and preference. For each cluster we create a virtual representative host, which is defined as the centroid of all the hosts in this cluster. The distance measure is carried out between any two clusters and computed as the distance of representatives of these two clusters. The clustering starts with each host being associated with a new cluster as the only member. All the distance values are normalized into the range $(0, 1)$.

3.2. Clustering Strategies

We use agglomerative algorithms for cluster formation. These algorithms initially place each host into a separate cluster and iteratively merge clusters until some stop criteria are met. The merging occurs in the following three steps: (1) Measure the distance between any two clusters and identify two clusters with the smallest distance as candidates for the next step. (2) Combine two candidates into a new cluster, and compute the representative host of this new cluster. The new cluster characteristics may be such that some hosts from the original two clusters become too distant from the new representative and are flagged as conflicts. (3) Conflicting hosts are expelled and a single-host cluster is formed for each such host. We compute the minimum distance between each cluster pair at the end of each iteration, and stop the clustering when this distance becomes larger than a treshold. The threshold value varies for different clustering applications.

4. Experiments and Applications

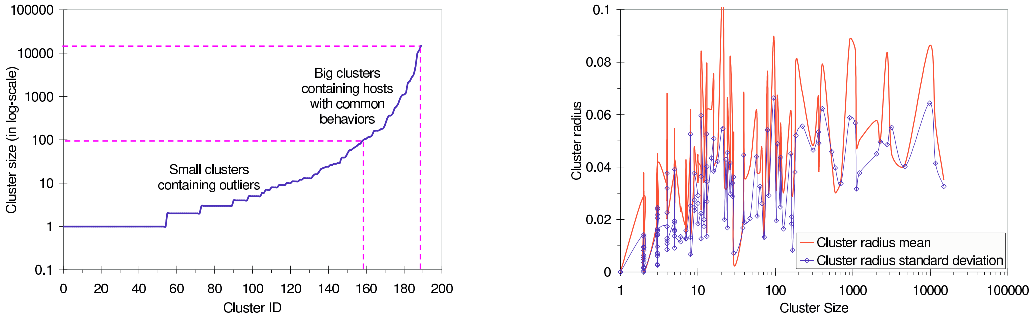
In this section we present some possible applications of host profiles and clusters, and illustrate them with experiments.

4.1. Clustering Hosts from the Internet Traces

In this experiment, we applied our host clustering approach on Auckland-VIII traffic traces set from NLANR-PMA [6]. This is a two-week GPS-synchronized IP header trace captured in December 2003 at the link between the University of Auckland and the rest of the Internet. We used data of the first ten days (Dec 02-11, 2003) for this experiment. After the filtering step, we were left with $62,187$ active and frequent hosts. We created profiles of these hosts and applied the agglomerative algorithm for clustering with the threshold value of 0.15 as the clustering stop criterion.

Figure 2 shows the clustering result with 189 derived clusters. We sort the clusters based on their size (number of hosts inside) and draw the distribution of cluster size in Figure 2(a). Out of the 189 identified clusters, 158 contain fewer than 100 hosts, with total of $1,460$ hosts falling into these small clusters. On the other hand, the top 10 clusters contain total of $53,587$ hosts with an average size of more than $5,000$ hosts per cluster. This indicates that Internet hosts exhibit similar behaviors. Manual examination of hosts in small clusters shows that they have some abnormal behaviors, such as a huge

[1] We record average TTL values in the host profile because they are useful for distinguishing between hosts, and help us discover IP addresses of Network Address Translation (NAT) boxes.



(a) Cluster size

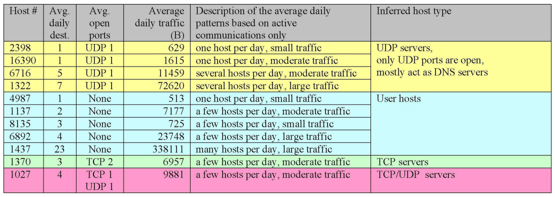
(b) Cluster radius

Fig. 2. Clustering result on trace of the first ten days from Auckland VIII data set with 189 clusters identified

volume of daily outgoing traffic to a small number of destinations which resembles a DoS attack pattern, or brief communication with a large number of destinations, which resembles scanning traffic. We expect that such small clusters with suspicious features will be present in any large traffic trace. They represent the anomaly of the daily Internet usage. On the other hand, more than 85% hosts fall into clusters larger than $1,000$, and represent a routine usage of the majority of the Internet hosts. We list the characteristics of these clusters in Figure 3.

We measure the quality of the clustering result by measuring the distance of each host from its cluster's centroid. Such a distance is called a *radius* of this cluster according to the host, and ranges from 0 to 1. A good cluster should have a low radius value for all the hosts inside, indicating high similarity between hosts. For each cluster, we compute the mean and standard deviation of host radius values, as indications of host intra-cluster similarity, and show them in Figure 2(b). The mean value is below 0.08 for most of the clusters, which indicates good concentration of members within clusters. Figure 2(b) also shows that the standard deviation does not promptly increase with the cluster size, so the similarity of hosts does not decrease with larger clusters.

To test our hypothesis that clustering of sampled backbone traces also produces useful data, we next applied our clustering technique to MAWI traces [7], collected at a trans-Pacific backbone link. The traces contain 15-minute long daily samples. We generated profiles using a three-day interval (Oct 19-21, 2005) and applied clustering to these profiles. The clustering produces results similar to the Auckland trace. We filtered about 86% of hosts in data preprocessing phase, and were left with 123,735 frequent and active hosts. The clustering produced 159 clusters, with top 10 clusters containing 94% of hosts. We will further investigate how to use sampled backbone traces for host profiling and anomaly detection in our future work.



Host #	Avg. daily dest.	Avg. open ports	Average daily traffic (B)	Description of the average daily patterns based on active communications only	Inferred host type
2398	1	UDP 1	629	one host per day, small traffic	UDP servers, only UDP ports are open, mostly act as DNS servers
16390	1	UDP 1	1615	one host per day, moderate traffic	
6716	5	UDP 1	11459	several hosts per day, moderate traffic	
1322	7	UDP 1	72620	several hosts per day, large traffic	
4987	1	None	513	one host per day, small traffic	User hosts
1137	2	None	7177	a few hosts per day, moderate traffic	
8135	3	None	725	a few hosts per day, small traffic	
6892	4	None	23748	a few hosts per day, large traffic	
1437	23	None	338111	many hosts per day, large traffic	
1370	3	TCP 2	6957	a few hosts per day, moderate traffic	TCP servers
1027	4	TCP 1 UDP 1	9881	a few hosts per day, moderate traffic	TCP/UDP servers

Fig. 3. Characteristics of clusters with more than 1,000 hosts

4.2. Evaluating Loyalty of Hosts to Clusters

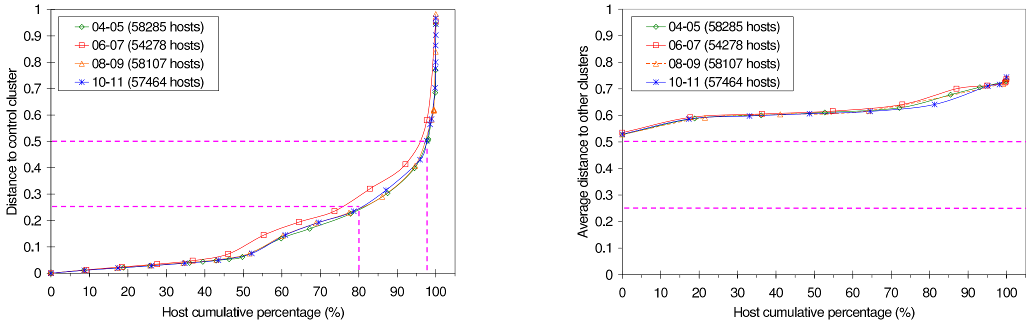
This experiment tests the hypothesis that legitimate hosts tend to fall into the same or a similar cluster, despite of their varying behavior over time. It is performed on the same data set as the previous experiment. Traces of two consecutive days are combined into a single trace prior to profiling and testing. We do this to increase the number of host profiles in the experiment, since many hosts appear once in two days but not every day.

To compare host behaviors with the characteristics of their belonging clusters, we first apply clustering on host profiles derived from the first two-day interval and tag each host with an ID of the cluster it belongs to. We call these clusters the "control clusters" for the corresponding hosts. We then use each remaining two-day interval to build host profiles based on it and for each host compute the distance between these profiles and the host's control cluster. The results of these tests are shown in Figure 4. For each test interval, more than 80% hosts have a distance lower than 0.25 to their control cluster. 98% hosts have such a distance of no more than 0.5. This result verifies the hypothesis that a large number of hosts exhibit steady behavior patterns over time. For each host, we also compute the average distance between its current profile and the clusters other than its control cluster, which reflects how dissimilar each host is from clusters other than its control cluster. Figure 4(b) shows that this average distance is always bigger than 0.5 for the four test intervals.

4.3. Applying Clustering for Slammer Detection

In this section we test if our host clustering and characterization approach can help detect suspicious changes in Internet traffic and thus give a timely alert about a potential Internet-wide security problem.

We use the Slammer trace data from NLANR-PMA, which was collected from all PMA monitors (all located



(a) Host distance to its cluster

(b) Host distance to other clusters

Fig. 4. Host loyalty, measured as distance from its own cluster and from other clusters.

on edge networks) on January 25-26, 2003, covering the period immediately before and during the outbreak of the Slammer worm. We distinguish traces collected before Slammer outbreak as those with no Slammer scans (UDP packets with 376-byte payload to port 1434), and use them to build host profiles. We then apply the clustering process on the host profiles and associate each host with a control cluster. In the experiment, we use the trace after the outbreak to build new host profiles and identify suspicious hosts by comparing new profiles with host control clusters. We build an Oracle to validate the correctness of our approach, by identifying each host that sends UDP packets to port 1434 with a 376 byte payload as infected. The distance of a new host's profile to the host's control cluster is shown in Figure 5 for infected and clean hosts. We use a threshold value of 0.25, as determined in previous section to separate normal from suspicious hosts. In Figure 5, nearly 90% of infected hosts have such distance larger than the threshold, and will be flagged as suspicious. This verifies our hypothesis that worm infection causes a sharp change in a host's behavior. When all hosts (both infected and clean) are observed, 28% of them have distance to their control clusters smaller than 0.25. This is clearly different from the 80% observed in the experiments shown in Figure 4(a), and signals an anomalous event. We conclude that host behavior changes can be used to indicate large-scale compromise of the Internet hosts and to identify majority of hosts with suspicious behavior. Note that 30% clean hosts also change their behaviors and have distance to their control clusters larger than the threshold. We manually examine the profiles of these hosts and find two main reasons for these false positives: (1) The training data trace is very short and some hosts appear rarely in this trace, resulting in poor profiles that do not model well these hosts' normal behaviors. (2) Due to the Slammer worm propagation and the conqeusent network congestion, hosts previously with a large volume of outgoing traffic reduce their sending rate which is recognized as a large change in their behaviors. With a sufficiently long training data and different scoring of lower versus higher host activity, we expect to reduce the false positive measure.

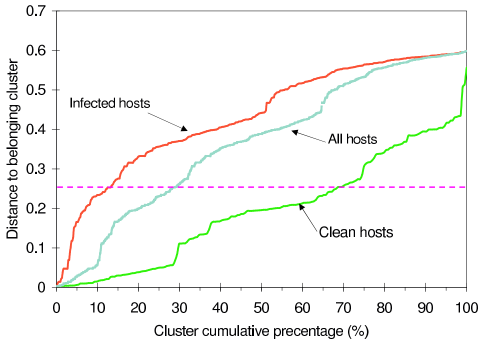


Fig. 5. Distance of infected and clean hosts from their control clusters, during the Slammer worm propagation.

5. Related Work

This paper applies data-mining to networking research in two steps: profiling hosts based on their behaviors and applying clustering techniques to categorize and characterize Internet hosts. Allman et al. [8] presents a distributed system for characterizing and sharing past behavioral patterns about network hosts. Instead of retrieving the behavior patterns from network traffic, they collect reports from network entities (e.g., host, subnet). Such design brings in trust problem, and it can not detect the anomalies instantaneously with online traffic.

Many researchers apply clustering to group hosts based on their relative positions [9][10][11]. to create clusters of hosts that are located close to one another. Our work applies clustering based on host behaviors instead of locations. Other researchers apply data mining (with techniques from statistics, machine learning, information retrieval, etc.) for anomaly and intrusion detection [12][13][14][15][16][17], with [16][17] based on host behaviors. Their host behavior profile consists only of the number of destinations contacted and a list of destination port numbers, with little consideration of individual communication patterns with specific peers.

Much research has focused on characterizing Internet traffic instead of hosts [18][19]. By processing the traces offline, flows are broken down into clusters with different characteristics. Since the Internet traffic varies broadly across different networks, these approaches either encounter performance challenges or produce unstable outputs for different traces. Instead of using raw trace traffic, [20][21][22][23][24] focus on using communication patterns or profiles of applications, with [22][23][24] using entropy to characterize traffic feature distributions. Compared with our work, [24] is most similar both in objectives and approaches. The authors build behavior profiles at host and service levels using source and destination IP addresses, port numbers and protocol field, and use entropy-based measure to define host categories. We build more detailed host profiles, that include communication and traffic volume statistics. This facilitates more precise characterization of a host's communication patterns. We further detect anomalies in a host's behavior by measuring how well this host follows its previously established behavior patterns.

There are also some commercial network defenses that are based on behavior modeling, with a goal of detecting and filtering anomalous network traffic. Mazu Enforcer [25] is a behavior-based network security system that monitors and models legitimate traffic patterns in the network, at fine-grain (hourly) basis. Peakflow platform [26] collects data from distributed network monitors and builds baseline models of normal network behavior. Our approach focuses on modeling individual host behaviors rather than one destination's network traffic, with the goal to detect the possible compromise and predict the future trustworthiness of a host.

6. Conclusion and Future Work

Understanding and characterizing typical host behaviors has important applications in the field of network security control. An accurate categorization of Internet hosts can help differentiate and identify malicious Internet hosts (and their users) from the mass of legitimate ones. In this paper, we discuss how to create host-behavior profiles based on Internet traffic traces, and how to use data mining and clustering techniques to automatically discover significant host groups based on created host profiles. Experiments with real Internet traces show that our profiling and clustering approach can derive host groups with significant features. We validate our hypothesis that the majority of Internet hosts tend to maintain same behavior patterns and fall into the same or similar groups over a moderately long time. We also demonstrate the applicability of our profiling and clustering approach to the detection of large-scale security incidents, using the Slammer worm spread.

Our future work will focus on using host profiles for building an Internet-wide host reputation system. We are also planning to apply our host clustering techniques to a wider range of Internet traffic traces, with the goal of building the models of Internet communication patterns. Such models are needed for a realistic simulation of Internet-wide events.

References

[1] Distributed Intrusion Detection System, *http://www.dshield.org/*

[2] V. Yegneswaran, P. Barford and S. Jha, *Global Intrusion Detection in the DOMINO Overlay System*, Proc. the Network and Distributed Security Symposium (NDSS) 2004.

[3] M. H. Dunham, *Data Mining Introduction and Advanced Topics*, Prentice Hall, 2003

[4] CAIDA CoralReef API,
http://www.caida.org/tools/measurement/coralreef/

[5] IANA port numbers,
http://www.iana.org/assignments/port-numbers

[6] NLANR PMA special traces archive, *http://pma.nlanr.net/Special*

[7] MAWI Traffic Archive, *http://tracer.csl.sony.co.jp/mawi/*

[8] M. Allman, E. Blanton and V. Paxson, *An Architecture for Developing Behavioral History*, Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), July 2005.

[9] B. Krishnamurthy, and J. Wang, *On Network-Aware Clustering of Web Clients*, Proc. ACM SIGCOM, August 2000.

[10] A. Agrawal, and H. Casanova, *Host Clustering in P2P and Global Computing Platforms*, Proc. Workshop on Global and Peer-to-Peer Computing on Large Scale Distributed Systems, May 2003.

[11] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker , *Topologically-Aware Overlay Construction and Server Selection*, Proc. INFOCOM 2002.

[12] D. Barbara, N. Wu, and S. Jajodia, *Detecting Novel Network Intrusions Using Bayes Estimators*, Proc. the First SIAM Conference on Data Mining, 2001.

[13] E. Bloedorn, et al., *Data Mining for Network Intrusion Detection: How to Get Started*, MITRE Technical Report, August 2001.

[14] W. Lee, and S. J. Stolfo, *Data Mining Approaches for Intrusion Detection*, Proc. the 1998 USENIX Security Symposium, 1998.

[15] S. Manganaris, M. Christensen, D. Serkle, and K. Hermix, *A Data Mining Analysis of RTID Alarms*, Proc. the 2nd International Workshop on Recent Advances in Intrusion Detection (RAID 99), September 1999.

[16] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan, *Data Mining for Network Intrusion Detection*, Proc. NSF Workshop on Next Generation Data Mining, November 2002.

[17] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, and V. Kumar, *A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection*, Proc. the Third SLAM Conference on Data Mining, May 2003.

[18] A. W. Moore and D. Zuev, *Internet Traffic Classification Using Bayesian Analysis Techniques*, Proc. the ACM SIGMETRICS, June 2005.

[19] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, *Flow Clustering Using Machine Learning Techniques*, Proc. the Passive & Active Measurement Workshop, April 2004.

[20] F.Hernandez-Campos, F. D. Smith, and K. Jeffay, *Statistical Clustering of Internet Communication Patterns*, Proc. the Symposium on the Interface of Computing Science and Statistics, 2003.

[21] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.Hu, *Behavior Profiling of Email*, Proc. the NSF/NIJ Symposium on Intelligence & Security Informatics, June 2003.

[22] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, *BLINC: Multilevel Traffic Classification in the Dark*, Proc. ACM SIGCOMM, August 2005.

[23] A. Lakhina, M. Crovella, and C. Diot, *Mining Anomalies Using Traffic Feature Distributions*, Proc. ACM SIGCOMM, August 2005.

[24] K. Xu, Z. Zhang, and S. Bhattacharyya, *Profiling Internet Backbone Traffic: Behavior Models and Applications*, Proc. ACM SIGCOMM, August 2005.

[25] Mazu Networks, *Mazu Enforcer*, *http://www.mazunetworks.com/*

[26] Arbor Networks, *Peakflow*, *http://www.arbornetworks.com*