PISA: Automatic Extraction of Traffic Signatures

Parminder Chhabra[1], Ajita John[2], and Huzur Saran[3]

[1] Winlab, Rutgers, The State University of New Jersey, NJ, USA
pchhabra@alumni. utexas.net
[2] Avaya Labs Research, Lincroft, NJ, USA
ajita@avaya.com
[3] Dept. of CS & Eng., Indian Institute of Technology, New Delhi, India
saran@cse.iitd.ernet.in

Abstract. Analysis of security attacks shows that an attack leaves its imprint or signature in the attack packets. Traffic from Distributed Denial of Service attacks and rapid worm spreads has the potential to yield signatures. While all signatures may not be indicative of attacks, it is useful to extract non-transient signatures that are carried by a sufficient number of flows/packets/bytes. The number of packets/bytes in the flows carrying the signature may be used for rate-limiting the flows, providing for timely and automated response to both known and unknown attacks. This paper proposes an efficient algorithm, PISA, which clusters flows based on similarity in packet information and extracts signatures from high-bandwidth clusters. Extensive experiments on two weeks of real attack data of 100 million packets yield about 1744 signatures. Additionally, PISA extracted the signature for the Blaster worm connection attempts in a mix of traffic from a trans-Pacific backbone link.

Keywords. Signatures, Traffic Clusters, Security, DDoS, Worms.

1 Introduction

Worms such as Code Red and Slammer [1] are spread by the repeated execution of similar pieces of malicious code. Distributed Denial of Service (DDoS) attacks are also typically caused by similar pieces of code executing on compromised hosts and launching network flows. The code exploits some vulnerability in the OS, an application or parts of the TCP/IP protocol and causes network systems to experience degradation in performance. When the same program is responsible for an attack, packets belonging to different flows of the attack may share the same values for several fields in the packets such as protocol header fields, application level fields and packet content. For example, in a SYN attack, the values for packet size, protocol, the SYN flag in the TCP header, destination address (if a particular server is being targeted), and destination port (if a particular kind of server is being targeted) are the same for all flows participating in the attack. The common values in the Code Red worm attack were in the following fields: *{message size, protocol, destination port}* [1]. Typically, unsolicited response traffic has common values for packet size and error type / code in ICMP packets and TCP flags in TCP packets. Some attacks launched by commonly used attack tools [7] have the same header checksum and

associated IP options. Additionally, similarity in application-level fields such as Subject, Body, and Attachment filenames can be found in SMTP-based email viruses.

Schemes that investigate packet header fields to protect against security attacks include [6], [9] and [3]. In [6], the authors propose a technique called Hop-Count Filtering to cluster destination address prefixes based on hop count values. Departures from baseline values of the TTL field in the IP header (hop count) are used to identify spoofed source addresses. A scheme for using RED-drop history as a way of identifying singleton high-bandwidth flows and aggregating these flows based on prefixes of destination IP addresses is proposed in [9]. Specialized techniques to aggregate flows based on specific fields (source address and port, destination address and port, and protocol) are proposed in [3]. Schemes that investigate packet content include [8] and [12]. In [8], the authors use a port-scan classifier to first identify suspected malicious flows. Next, they use string matches on packet content using Rabin fingerprints together with protocol and destination port as a way of identifying worm signatures. In [12], the authors use string matches on packet content together with destination port as a basis for identifying worm signatures. The above schemes are effective for the specific fields they target. However, there exists a need to study the problem of aggregating traffic using any subset of fields because, as known attacks have shown and unknown attacks may possibly show, characteristics of attacks may extend to other fields. Investigating any subset of fields may lead to the automatic generation of generalized attack profiles that may form the basis for an automated response to unknown attacks. There are, of course, varying degrees of cost for retrieving values for different fields in a packet. However, as the severity of attacks continue to increase, this cost may be justified.

The work described in this paper addresses the problem of finding high bandwidth aggregates of flows that share similar values for any subset of fields in their packets. This subset of fields and their values form *signatures*. While all signatures may not be indicative of attacks, signatures that are carried by a large number of flows/packets/bytes and occur frequently in traffic are of high interest to network administrators for early detection of possible malicious activity. The paper discusses packet signatures and their relevant properties. The paper discusses the Packet Imprint in Security Attacks algorithm (PISA). PISA samples incoming packets at a network element and groups the flows corresponding to the packets into clusters based on similar values in the fields of the packets. A cluster is intensive if it consumes a large proportion of the bandwidth as measured by the number of packets/bytes across all flows in the cluster. PISA extracts the signatures of intensive clusters from samples of data over time and filters out transient signatures. The signatures carry associated information such as the number of samples that the signatures appeared in (persistence) and the average number of flows (distribution) and packets/bytes (intensity) that belong to the signature. PISA provides an aggregated view of the traffic that arrives at a network element and the aggregation is not constrained to any particular field in the packet. The effectiveness of our approach is shown through experiments with an implementation of PISA on real network data, which demonstrate that PISA can extract signatures of high value such as those that correspond to connection attempts in a worm spread. Additionally, PISA was able to aggregate large amounts of real attack data to yield signatures that showed distribution of the attacks over a variety of fields.

The key contributions of our work are as follows: We define characteristic properties of signatures – Dimension, Intensity, Persistence and Distribution. We propose PISA, a randomized non-exponential algorithm for clustering traffic flows based on any subset of fields in packets to yield signatures. Signatures may carry any combination of fields that may be defined for a packet. No pre-computed signatures are required. Only a sample of traffic is required to extract significant signatures. In the face of extreme performance degradation at a network element, PISA can be used as a basis for self-healing of the network element by penalizing traffic that consumes the most resource(s).

The paper is organized as follows. Section 2 discusses signatures and clusters of flows. Section 3 describes PISA. Section 4 discusses experiments and results on traffic traces. Finally, conclusions are presented in Section 5.

2   Signatures from Clusters of Network Flows

Definition: A signature is given by k ordered pairs $\{(f_1, v_1), (f_2, v_2), \dots (f_k, v_k)\}$, where $k \geq 1$ is the dimension of the signature, f_i is a field in a packet and v_i is a value for f_i, $1 \leq i \leq k$.

A signature S is a subset of another signature T if the ordered pairs in S are contained in T. A flow is said to *carry* a signature if the ordered pairs in the signature match with values for the corresponding fields in the packets of the flow. The notion of a flow used in this work is a 5-tuple with the following fields: (source address, source port, destination address, destination port, protocol). An example of a signature is as follows: *{(packet_size, 48), (src_port, 80), (src_addr, 10.0.0.1), (type, tcp), (tcp_flag, Syn ack), (dest_port, 3072)}*. The signature is carried by flows originating from source 10.0.0.1 and port 80 with SYN ack packets of size 48 bytes to port 3072 of destinations. Four properties that characterize signatures are defined as follows:

1. Dimension: The dimension of a signature is the number of field-value pairs in it. A higher dimensional signature contains more information about the type of flows carrying the signature. A signature may be useful only if it contains a minimum number of fields in it.

2. Intensity: The intensity of a signature is the average number of packets/bytes that carry the signature. Intensity reflects the bandwidth consumption of flows carrying the signature.

3. Persistence: The persistence of a signature is its activity over time. It may be represented as a history of the frequency of occurrence of a signature over a time interval.

4. Distribution: The distribution of a signature is the average number of flows (alternatively, prefix matches of IP addresses across flows) carrying the signature and is useful in determining the spread of the signature.

The *significance* of a signature can be defined in terms of its dimensionality, intensity, persistence, and distribution. For example, a system may define significant signatures to be those that contain at least 4 fields, have a bandwidth consumption of

at least 2% (intensive), appear for at least 60 seconds (persistent), and be carried by 5% of the traffic flows.

Definition: An m-dimensional cluster C_m is defined as a set of flows in which all flows have similar values for each field in a set of m fields. While exact matches are discussed in this paper as the basis of similarity, the approach can be extended to include approximate matches. The m fields form a signature carried by all flows in the cluster.

The set of flows in a network sample can be grouped into *clusters* of flows such that all flows in a cluster carry a "maximal" signature, where the maximal signature for a set of flows is the largest set of similar field-value pairs. A flow can belong to more than one cluster. Note the following properties:

- An m-dimensional cluster is also an n-dimensional cluster where $1 \leq n \leq m$.
- For two clusters X,Y with signatures S_x, S_y, $X \subset Y$, ➔ $S_y \subseteq S_x$ …….**(a)**

Each flow in a cluster has a weight given by the number of packets/bytes in that flow. The weight of a cluster can be defined as the sum total of the weights of all the flows in that cluster as follows: Weight(cluster C) = Sum of weights of all flows in C ….. **(b)**

The weight of a cluster is indicative of the bandwidth it consumes. An *intensive* cluster is one whose weight is above a specified threshold. *Intensive clusters yield intensive signatures.*

3   PISA: Algorithm

There has been prior work that has looked at identifying singleton flows that are intensive [9] [15]. The motivation in this work is to efficiently extract high-dimensional, intensive, persistent and distributed signatures in network traffic. The approach is to scan a sample of the incoming traffic at a network element and group the flows in that sample into intensive clusters, which yield intensive signatures. Different samples of data (closely spaced in time) yield a history of intensive signatures. Transient signatures can be pruned out from this history. The non-transient signatures along with related information such as the number of scans in which the signature was seen (persistence), the average number of flows (distribution) and the average number of packets/bytes (intensity) carrying each signature provides a history of aggregated traffic at the network element.

Clustering techniques can be broadly divided into hierarchical and partitional [2] [5]. Hierarchical clustering techniques consist of successively combining smaller clusters into larger ones or by splitting larger clusters into smaller ones. Partitional clustering techniques decompose the data into a set of disjoint clusters based on the optimization of a criterion function. Exhaustive techniques to search for all clusters in a data sample using any of the techniques discussed above are expensive [10]. We argue that exhaustive techniques are not necessary in this work because persistent and distributed signatures should be easily extracted in approximation-based methods. PISA repeatedly executes over samples of data across time intervals. Hence, it is not necessary to find all intensive signatures in an execution, as persistent signatures that are missed in one execution will likely show up in subsequent ones. Randomized approaches [4] allow faster extraction of clusters than exhaustive approaches. PISA

uses a hierarchical and randomized clustering technique on a lattice structure (which is commonly used in clustering methods) of the sampled flows.

The parameters in PISA are as follows: (i) Sample size or the number of packets sampled in each scan phase: N. (ii) The minimum desired dimension of a signature: k (iii) Threshold for an intensive cluster: T. (iv) The fields of interest in the sampled packets. PISA consists of two parts: (1) A scan phase that extracts a sample of the network traffic (2) A signature extraction phase that clusters the flows in the sample to generate signatures.

In the scan phase, PISA samples a specified number of packets from the packets arriving at a network element. It maps the sampled packets to network flows and populates a *flow table* where each entry contains a flow, the number of packets belonging to the flow and the fields of interest in the packets and their corresponding values.

In the signature extraction phase, the flows in the flow table are grouped into clusters. Let F be the set of flows in the flow table. Consider a lattice of all possible subsets of F and referred to as the flow lattice in this paper. The empty set ϕ is at the bottom of the lattice and F is at the top of the lattice. An edge exists between two subsets S_1 and S_2 if S_1 is a superset of S_2 and $|S_1| = |S_2| + 1$. The lattice consists of $|F|+1$ levels where each level i has all the subsets of F with cardinality i (See Fig.1). The intensive clusters of F are points on the lattice. The goal is to find intensive signatures of dimension greater than or equal to k.
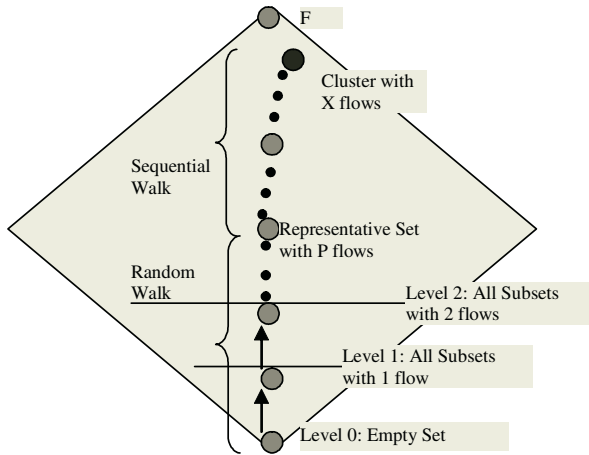

Fig.1. Lattice of Subsets of Flows and Signature Extraction

Let h be the number of fields of interest in the packets and k $(1 \leq k \leq h)$ be the minimum number of fields desired to be similar in a cluster. Consider a path $(\phi, S_1, S_2 \dots, F)$ from the bottom of the lattice to its top. Trivially, ϕ and S_1 (consisting of a single flow) are h-dimensional clusters. Also, if S is a superset of P and S is an i-dimensional cluster and P is a j-dimensional cluster, then $i \leq j$ (follows from property (a) in Section 2). Thus, the dimension of a cluster (and the dimension of its associated

signature) monotonically decreases up the lattice. This implies that for any $1 \leq k \leq h$, either F is a k-dimensional cluster or there must exist two subsets of F, S_t, S_{t+1}, such that S_t is an i-dimensional cluster, S_{t+1} is a j-dimensional cluster, there is an edge between S_t, and S_{t+1}, and $j < k \leq i$. S_t is called a k-dimensional representative set. Along the path from the bottom of the lattice to S_t, there may exist several representative sets of dimensions greater than k. The number of paths to a representative set is exponential in the number of flows in the representative set.

The signature extraction phase in PISA attempts to find a path to a representative set by randomly picking flows to move up the levels of the flow lattice. It starts at the bottom of the lattice by initializing a set S as the empty set ϕ. It traverses up the lattice in a sequence of steps. In each step it randomly chooses a flow from the flow table to add to the set S, each time checking if the resulting set is a representative set of dimension k or greater. It collects all such representative sets. It stops the traversal when the resultant set is a cluster of dimension less than k. This part of the walk will be called the random walk. A cluster forms the top of a sub-lattice that contains all subsets of the cluster. The more flows there are in the cluster, the larger is this sub-lattice and the greater the probability of remaining within the sub-lattice during the random walk and finding a representative set for the cluster.

At the end of the random walk, each collected representative set is expanded into a cluster by matching the headers of each flow in the flow table against those of the representative set. This corresponds to walking the sub-lattice corresponding to the cluster by sequentially searching the flow table. This part of the walk is called the sequential walk. The set resulting from the growth of the representative set is a cluster of flows that share common values for at least k fields. This cluster will yield an intensive signature if the weight of the cluster is greater than the specified threshold T.

Each walk through the lattice (random + sequential) will be referred to as a lattice traversal. Lattice traversals can be repeatedly executed to discover an increasing number of the intensive clusters in the lattice. The limit on the number of lattice traversals can be set through a parameter as discussed later. Note that PISA does not construct the entire lattice. It constructs sets of flows as it encounters them along the lattice traversals. PISA adds the signatures of discovered intensive clusters to a *signatures table*, which stores the fields and values for each signature. Additional information for each signature is the number of samples in which it was seen, the average number of flows and packets/bytes carrying the signature. A Time To Live (TTL) field associated with the signatures flushes out signatures not seen over many samples. A signature can be logged before it is flushed out.

Consider the four parameters in PISA: N, T, k and the fields of interest in the packets. N is the number of packets sampled in the scan phase. Larger values of N will typically increase the size of the flow lattice and may yield more signatures. However, reasonably sized values for N should be sufficient for finding intensive and persistent signatures. This claim is supported by experiments presented in Section 4. T (threshold for the weight of an intensive cluster) can be specified as a percentage of the traffic. Low values of T may yield many signatures and high values may yield fewer signatures. k is the minimum dimension of a signature. If it is set to low values such as 1 or 2, the number of detected signatures may be very high and many of them may be meaningless. If k is set to too high a value, few or no signatures may be

detected. The fields of interest in a packet may include protocol header fields, application level fields and fields that may be defined on the packet content.

Two approaches to determine the number of lattice traversals *I* are: (1) Static: *I=L*N*, where *L* is a constant called the lattice factor (2) Dynamic: Use a constant called the exit factor, which is the maximum number of consecutive lattice traversals where no new intensive clusters (or, a few) are seen. Since signatures that are carried by a large number of flows should have a large number of representative sets, they should be found quickly and it may be necessary to traverse only a small percentage of the paths in the lattice to detect them.

3.1 Analysis of PISA

Assume that the number of packets in a scan is *N*, the number of flows in a scan is *f*, and the number of fields of interest is *h*. In each lattice traversal, representative sets are collected and each representative set is grown against the flow table. Any path in the lattice is of length *f*. At every node in the path taken by a lattice traversal, at most *h* comparisons are made between fields of the cluster at the previous node in the path and the newly added flow. The maximum number of representative sets along a path is *(h-k+1)*. The time taken to grow a representative set is proportional to *f*. If there are *I* lattice traversals, the upper bound on the time taken for PISA is O*(I * (f * h + (h-k+1) * f * h))*. Since *h*, *k* are constants and *N* is an upper bound on *f*, the time complexity can be expressed as O(*I*N*). If the lattice factor is used to determine *I*, the time complexity is O(N^2). If the exit factor is used, *I* is directly proportional to the number of intensive clusters in the sample and the time complexity of the algorithm is O(Number of intensive clusters * *N*).

The space requirements of the algorithm are dominated by the size of the flow table: O(*N*) and the size of the signatures table: O(Number of Signatures*k). *k* is a small constant. The number of signatures can be controlled through *T* and the TTL for each signature.

PISA does not report *any* signature that does not belong to a cluster of flows in a sample. Hence, there are no false positives. The probability of finding a signature S in a sample is proportional to the number of flows carrying the signature. This can be estimated, but is not presented here for brevity. The likelihood of missing signatures that are carried by a large number of flows is low.

4   Experimental Setups and Results

An implementation of PISA, referred to as the PISA system, repeatedly executes the scan and signature extraction phases. Extracted signatures are stored in the table of signatures that are logged after a fixed number of executions, referred to as a *scan set*. In our experiments, there are 500 executions in a scan set. All signatures fall into two categories: transient and non-transient. A signature is *transient* if it occurs in fewer than 50 scan phases at the end of each scan set. Non-transient signatures can be divided into two categories - persistent and non-persistent. A signature is *persistent* if it occurs in at least 500 scan phases. A signature is *non-persistent* if it is non-transient and occurs in less than 500 scan phases. The *frequency* of a signature is a measure of

its occurrence in a scan set. The results reported in this paper consider non-transient signatures only. The value for the TTL field was 500. Significant signatures are defined to be those that have at least 4 dimensions (dimensionality), have a bandwidth consumption of at least 10% (intensity), and occur for at least 500 scans (persistence).

Two kinds of data were used in our experiments[1]: (1) Traces from CAIDA [14] provided a testbed of DDoS attacks and (2) data from the WIDE backbone [13] in the week of Blaster worm outbreak served as a testbed to identify worm spread signature in a mix of traffic.

4.1 CAIDA Trace Data and Results

CAIDA traces contain a wide variety and number (~8000) of attacks and have been analyzed by independent measurements [11]. CAIDA traces are useful in understanding the depth of information that can be captured by the PISA system and served as a useful test-bed to reveal different properties of PISA. The CAIDA trace used in our experiments consisted primarily of backscatter data from nearly two weeks (denoted by Week-I-II). *Backscatter packets* are defined as unsolicited response packets that include ICMP (host unreachable, port unreachable, time exceeded etc.) or TCP packets (syn-ack and reset flows). Details of the experimental setup to capture traces on the /8 network can be obtained in [11]. Since there is a strong correlation between attack packets and backscatter packets, PISA applied to backscatter packets extracts signatures that can be correlated to the signatures for attack flows. For example, if a signature for backscatter flows contains the value SYN-ACK for the TCP flag, the signature in the attack flows will contain the value SYN for the same TCP flag.

The values of the parameters in our experiments (unless otherwise specified) are as follows: $k = 4$, $T=10\%$ of N, $L = 20$, $N = 50$. TCP/ICMP/UDP and IP header fields are considered for inclusion in packet signatures. The criterion for transient signatures (50 scans) in the PISA system approximately corresponds to the criteria (attacks lasting for less than a minute) for signatures ignored in [11]. Unless stated otherwise, only persistent signatures were considered for analysis. The purpose of the experiments was to study (a) the effect of different parameters on PISA. (b) the total number, duration and type of attacks observed over Week-I-II. In (a), the use of several values for each parameter in experiments necessitated working with a smaller data sample than Week-I-II. A 21.6 hour long data sample was chosen without bias. This is a sufficiently large data sample to mitigate short-term perturbations in traffic.

To study the effect of varying the number of lattice traversals, I was set to a constant (lattice factor) times the number of flows N. Fig. 2 compares the total number of distinct signatures when lattice factors of 5, 10, 20, and 50 are used for a sample size of 50. Similar results were obtained for sample sizes of 25, 40 and 75 packets. The total number of signatures does not increase significantly for lattice factors of 10, 20 or 50, showing the low rate of false negatives. While the number of signatures is reported here, the actual signatures too did not vary as the lattice factor

---

[1] The authors acknowledge the Senior Staff at CAIDA for the availability of the traces [14] from their work [11] and MAWI working group [13] for making traces from the WIDE backbone available to us.

was increased. An analysis of the signatures showed that frequency of occurrence of persistent signatures (not shown here) did not change significantly (low rate of false negatives) as the lattice factor varied between 5-50. For example, as the lattice factor is varied between values of 5-50, the frequencies of occurrence of a highly persistent signature were observed to vary between 94-96%. The frequencies of occurrence of another lesser persistent signature were observed to vary between 66-77%, again demonstrating the low rate of false negatives for persistent signatures.
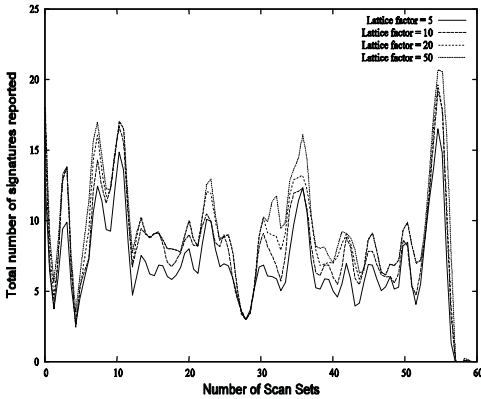

Fig. 2. Number of signatures for various lattice factors

Table 1. Frequency of non-persistent signatures as a percentage of scan set (500 scans) for varying N

Frequency →	10-25%	25-50%	50-75%	>75%
N=10pkts	24	1	1	0
N=25pkts	39	1	1	1
N=50pkts	53	1	1	1
N=75pkts	53	1	1	1

Table 2. Variation in the number of persistent signatures with different values of threshold, T

Threshold Value of T	Number of persistent signatures
5%	33
10%	28
20%	24

Table 3. Distribution of signatures by TCP and ICMP Protocols

Sig. Type	Total	50-120 scans	120-500 scans	500-10K scans	>10K scans
TCP	1210	283	661	192	19
ICMP	534	250	12	197	30
Total	1744	533	673	489	49

Table 4. Distribution of Signatures by response protocol

Protocol Type	Number of signatures
TCP (total)	321
TCP (RST, RST ACK)	273
TCP SYN ACK	31
TCP Other	17
ICMP (total)	110
Host Unreachable	57
TTL Exceeded	51
Parameter Problem	1
UDP	1

Table1 shows the number of non-persistent signatures after every scan set. The signatures that occurred in more than 25% of the scans did not change significantly as N varied. As N increases, the number of signatures that occur between 10-25% of the scans increases but is the same for N=50, 75. An examination of the signatures showed that the same signatures are extracted as N varies between 10-75. Similar

results were observed for persistent signatures. The results support our earlier assertion that N need not be very large to extract the intensive signatures. In Section 4.2, results using higher values of N are presented.

Table 2 shows how T can be used to control the number of signatures. As the value of T increases from 5% to 20%, the number of signatures decreases since fewer clusters of flows will have a weight greater than T.

Having shown the effect of varying values of parameters in PISA, we present results from the execution of PISA on Week-I-II.

Signatures of dimensions between 4-6 were detected in our experiments. About 65% of the signatures are 4-dimensional, about 25% are 5-dimensional and about 10% of the signatures are 6-dimensional. Consider a source sending SYN requests to multiple destinations of which two destinations are prominent. Sample observed signatures are presented below. The values for each field are represented by *a*, *b*, *c*, *d*, *x*, and *y*.

$S(5)$: 5-dimensional, *{(packet_size, a), (src_port, b), (src_addr, c), (type, d), (tcp_flag, SYN)}*

$S(4)$: 4-dimensional, *{(src_port, b), (src_addr, c), (type, d), (dest_addr_1, x)}*

$S_1(6)$: 6-dimensional, *{(packet_size, a), (src_port, b), (src_addr, c), (type, d), (tcp_flag, SYN), (dest_addr_1, x)}*

$S_2(6)$: 6-dimensional, *{(packet_size, a), (src_port, b), (src_addr, c), (type, d), (tcp_flag, SYN), (dest_addr_2, y)}*

$S(4) \subset S_1(6)$ and $S(5) \subset S_1(6)$ and $S(5) \subset S_2(6)$. A comparison of signatures shows: (1) $S(4)$ occurs very infrequently and contains a slightly larger number of flows compared to $S_1(6)$. $S(5)$ occurs more frequently than $S_1(6)$ and $S_2(6)$. (2) The average number of flows / packets in $S(5)$ is more than twice that of either $S_1(6)$ or $S_2(6)$.

For reasons of ease, signatures with the protocol field value of TCP/ICMP will be referred to as TCP/ICMP signatures, respectively. Table 3 shows the frequency of TCP and ICMP signatures. Signatures occurring between 50-500 scans (non-persistent) correspond to attack durations of about 2-20 minutes. Signatures that occur between 500-10,000 scans correspond to attacks that last between 20 minutes-several hours. Signatures occurring in more than 10,000 scans correspond to attacks lasting over several hours and up to several days. The results are consistent with [11], which states that 50% of the attacks are less than 10 minutes in duration, 80% are less than 30 minutes, 90% last less than an hour and the rest span from several hours to a few days.

Distribution of signatures by response protocol is presented in Table 4. ICMP signatures with the error code "TTL Exceeded" occur in about 50% of all ICMP signatures. About 80% of TCP signatures contain the RST/RST ACK flag in the TCP flag field. The total number of TCP signatures is about three times the number of ICMP signatures. This agrees with attacks classified on the response protocol in [11].

Fig. 3 shows plots for the total number of persistent signatures and the number of persistent TCP signatures. ICMP signatures make up the difference between the two plots.

It is difficult to directly compare the number of signatures from the PISA system to the ones reported in [11] because the approach in [11] models the attacks as emanating from groups of flows where each group contains flows all destined to the same IP address. Thus, similar flows to different destination IP addresses are not

classified into the same attack. An examination of the signatures from our experiments shows that 85% of the signatures collected by the PISA system did not have destination IP address as a field in the signature because PISA clustered flows to different destinations based on other fields in the packets. This explains why the signatures identified by PISA is about one-fourth of the number seen in [11]. This is significant because PISA enables multiple simultaneous attacks that share similar characteristics (in fields) to be represented by a single signature.

Results may be summarized as follows: (1) 100 million packets in Week-I-II yielded about 1744 signatures. (2) The threshold (T) controls the number of intensive signatures. (3) A variety of fields appear in the signatures including protocol, packet size, source address, source port, ICMP error types, TCP flags, destination address and destination port. (4) Intensive signatures can be extracted from small samples of data, thus enabling signatures to be detected quickly. (5) A large number of lattice traversals are not required to detect signatures. (6) Signatures with dimensions varying from 4-6 were extracted from the data. This shows that varying levels of information may be gathered in attacks. (7) The duration and the type of signatures are comparable to those reported in [11].
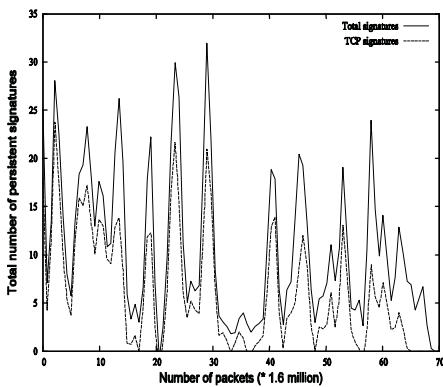

Fig. 3. Total number of persistent signatures and number of persistent TCP signatures every 1.6 million packets
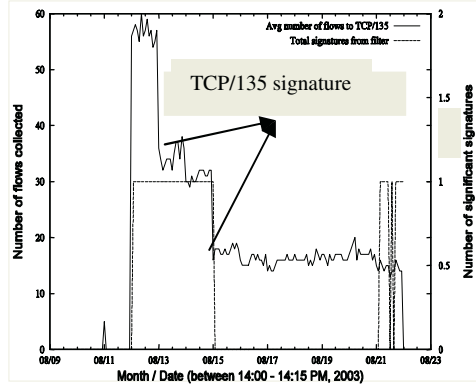
Fig. 4. Persistence (secondary y-axis) & average number of flows (primary y-axis) of the signature of Blaster connection request to TCP/135

4.2 Experience with the Blaster Worm

This section discusses the effect of executing PISA on traffic gathered in the week of the Blaster [1] worm outbreak. Details of the Blaster worm spread and attack phase can be found in [1]. While the worm was active on multiple ports, TCP/port 135 is considered for brevity. Traffic collected during a 15-minute intervals for 2 weeks (9-21 Aug 2003) from the WIDE [13] backbone are analyzed. The attack traffic contributed less than 5% of the bytes over the measured interval.

The following values were used for the PISA parameters: T=9%, k=4. About 25% of the traffic was sampled, which corresponded to 120-250 packets per scan.

Fig. 4 shows the number of significant signatures per scan set and the average number of flows carrying the signature corresponding to attempts to connect to TCP port 135. The connection attempt, that shows up as a persistent and intensive signature was identified as *{(packet_size, 48), (dest_port, 135), (type, TCP), (tcp_flag, SYN)}*. Non-persistent signatures that are supersets of this persistent signature and of the form *{(packet_size, 48), (dest_port 135), (type, TCP), (tcp_flag, SYN), (src_addr, [IP1, IP2 … IPn])}*, where IP1, IP2 … IPn denote different IP addresses, were observed frequently. The figure also shows persistent signatures on 21 and 22 August for RSTP data transfers and ICMP/92 (echo requests with a packet size of 92 bytes). ICMP/92 corresponds to the ping sweep of the Welchia.A [1] worm. Other non-persistent signatures observed corresponded to NNTP and FTP data transfer, and connection requests to web servers. When *T* was reduced to 7%, additional persistent signatures corresponding to RSTP, connect requests to web servers and ICMP/92 (from 19 August onwards) were observed.

5   Conclusions

Signatures represent aggregated characteristics of network flows that is the hallmark of many DDoS attacks and worm spreads. This paper defines characteristic properties of signatures such as dimensionality, intensity, persistence, and distribution, which help define significant signatures that may be of interest while analyzing network traffic. The paper proposes PISA, an efficient algorithm to extract significant signatures from samples of traffic. The fields in the signatures enable filters to be designed to isolate flows suspected of belonging to attacks and the persistence and intensity of signatures can form the basis for rate-limiting the isolated flows. The paper presents experimental results for real network traces that show that PISA works very effectively in extracting signatures for a wide range of attack scenarios. PISA abstracts about 100 million packets participating in denial of service attacks to 1,744 significant signatures. In addition, the results show that PISA effectively extracts the signature for the connection requests corresponding to the Blaster worm in a mix of traffic from a backbone link.

References

[1] CERT, Vulnerabilities, Incidents and Fixes, http://www.cert.org/nav/index_red.html

[2] R. O. Duda and P. E. Hard. Pattern Classification and Scene Analysis. Wiley-Interscience, NY, 1973.

[3] C. Estan, S. Savage and G. Varghese, "Automatically Inferring Patterns of Resource Consumption in Network Traffic", *Proceedings of the ACM SIGCOMM Conference*, Karlsruhe, Germany, August 2003.

[4] H. V. Jagadish, J. Madar, and R.T. Ng, "Semantic Compression and Pattern Extraction with Fascicles", *Proceedings of 25th VLDB*, pp. 186-198, 1999.

[5] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, New Jersey, 1988.

[6]   C. Jin, H. Wang, and K. G. Shin. "Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic", *ACM Conference on Computer and Communications Security (CCS)'2003* , October 2003.

[7]   D.   Dittrich,   "Distributed   Denial   of   Service   Attacks/Tools", http://staff.washington.edu/dittrich/

[8]   H.-A. Kim and B. Karp, "Autograph: Toward Automated, Distributed Worm Signature Detection", *Proceedings of the 13th Usenix Security Symposium (Security 2004)*, San Diego, CA, August, 2004.

[9]   R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling High Bandwidth Aggregates in the Network." *Computer Communications Review* 32:3, July 2002, pp. 62-73.

[10] H. Mannila and H. Toivonen. Level_Wise search and borders of theories in knowledge discovery, *Data Mining and Knowledge* Discovery, 1,3, pp 241-258.

[11] D. Moore, G. Voelker, and S. Savage, "Inferring Internet Denial of Service Activity", *Proceedings of the 2001 USENIX Security Symposium, Washington D.C., August 2001.*

[12] S. Singh, C. Estan, G. Varghese, and S Savage, "Automated Worm Fingerprinting", *Proceedings of the 6th ACM/USENIX OSDI Symposium*, San Francisco, CA, December 2004.

[13] MAWI   Working   Group,   "Packet   traces   from   WIDE   backbone   ", http://tracer.csl.sony.co.jp/mawi/

[14] UCSD Network Telescope Backscatter Datasets for February 2001, CAIDA, http://www.caida.org/analysis/security/telescope/

[15] P. Chhabra et. al. XCHOKe: Malicious Source Control for Congestion Avoidance at Internet Gateways. Proceedings of 10th IEEE ICNP, 2002.