Knowledge-Independent Traffic Monitoring: Unsupervised Detection of Network Attacks

Pedro Casas[1,2,3], Johan Mazel[1,2], and Philippe Owezarski[1,2]
[1]CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
[3]Corresponding author. Telephone: +33 (0)5 61 33 68 05 - Fax: +33 (0)5 61 33 64 11
Email: {pcasashe, jmazel, owe}@laas.fr

***Abstract*—The philosophy of traffic monitoring for detection of network attacks is based on an "acquired knowledge" perspective: current techniques detect either the well-known attacks which they are programmed to alert on, or those anomalous events that deviate from a known normal-operation profile. These approaches rely on an expert system which provides the required knowledge, either in terms of "signatures" of the well-known attacks or as anomaly-free traffic datasets, rich enough to build representative profiles for normal-operation traffic. In this article we discuss the limitations of current knowledge-based strategy to detect network attacks in an increasingly complex and evolving Internet, characterized by ever-emerging applications and an ever-increasing number of new network attacks. In a diametrically opposite perspective, we place the emphasis on the development of unsupervised detection methods, capable of detecting unknown network attacks in a dynamic environment without any previous knowledge, neither on the characteristics of the attack nor on the baseline-traffic behavior. Based on the observation that a large fraction of network attacks are contained in a small fraction of traffic flows, we demonstrate how to combine simple clustering techniques to accurately identify and characterize malicious flows. To show the feasibility of such a knowledge-independent approach, we develop a robust multi-clustering-based detection method and evaluate its ability to detect and characterize network attacks without any previous knowledge, using packet traces from two real operational networks.**

***Index Terms*—Unsupervised Detection of Attacks, DDoS, Network Scans, Robust Clustering, Automatic Characterization.**

I. Introduction

NETWORK traffic monitoring has become an essential means for detection of network attacks in today's Internet. The principal challenge in detecting network attacks is that these are a moving target. It is not possible to know the different attacks that an attacker may launch, because new attacks as well as new variants of already known attacks are continuously emerging. Indeed, attacks have become both increasingly numerous and sophisticated over the years [1].

Two different approaches are by far dominant in current research community and commercial detection systems: signature-based detection and anomaly detection. Despite being opposite in nature, both approaches share a common downside: they rely on the knowledge provided by an expert system, usually a human expert, to do the job. We shall therefore refer to them as knowledge-based detection approaches.

On the one hand, signature-based detection systems [2] are based on a extensive knowledge of the particular characteristics of each attack, referred to as its "signature". Such systems are highly effective to detect those well-known attacks which they are programmed to alert on. However, they cannot defend the network against new attacks, simply because they cannot recognize what they do not know. In addition, building new signatures involves manual inspection by human experts, which is not only very expensive and prone to errors, but also introduces an important latency between the discovery of a new attack and the construction of its signature. In a network scenario where new attacks are constantly appearing, such a manual process imposes a serious bottleneck on the defense capabilities of the network.

On the other hand, anomaly detection [3]–[7] relies on the existence of normal-operation traffic instances to build a baseline-profile, detecting anomalies as traffic activities that deviate from it. Such an approach permits to detect new kinds of network attacks not seen before, because these will naturally deviate from the constructed baseline. Nevertheless, anomaly detection requires training to construct normal-operation profiles, which is time-consuming and depends on the availability of purely anomaly-free traffic datasets. Labeling traffic as anomaly-free is expensive and hard to achieve in the practice, since it is difficult to guarantee that no anomalies are hidden inside the collected traffic. Additionally, it is not easy to maintain an accurate and up-to-date normal-operation profile, particularly in a dynamic and evolving context where new services and applications are constantly emerging.

Motivated by the limitations of knowledge-based approaches, a new research area has emerged in the last years, based on a diametrically opposite philosophy for detection of anomalous traffic events: Unsupervised Anomaly Detection. Instead of relying on a previously acquired knowledge on the characteristics of network attacks or on the baseline-traffic behavior, unsupervised detection uses data-mining techniques to extract patterns and uncover similar structures "hidden" in unlabeled traffic of unknown nature (attack or normal-operation traffic). Based on the observation that network attacks, and particularly the most difficult ones to detect, are contained in a small fraction of traffic flows w.r.t. normal-operation traffic [8], their unsupervised detection basically consists in identifying “outliers”, i.e. patterns that are distant from the majority of the traffic.

Some methods for unsupervised detection of network attacks have been proposed in the past [9]–[13]; the majority of them are based on clustering techniques and outliers detection. The objective of clustering is to partition a set of unlabeled elements into homogeneous groups of “similar” characteristics, based on some similarity measure. Different from other techniques for unsupervised data analysis (e.g. density estimation, dimensionality reduction, etc.), clustering permits to work with multiple-classes problems without modifying the characteristics of the analyzed traffic, hence it represents an attractive means for unsupervised detection of attacks. Unfortunately, even if hundreds of clustering algorithms exist [17], it is very difficult to decide which algorithm would be the best one for our particular problem. Different clustering algorithms produce different partitions of data, and even the same clustering algorithm provides different results when using different initializations and/or different algorithm parameters. This is in fact one of the major drawbacks in current cluster analysis techniques: the lack of robustness.

In this article we stress the paramount advantage of unsupervised, knowledge-independent detection algorithms based on clustering, but we argue that their performance should not be tied to the particular characteristics of any clustering algorithm. We shall therefore present an alternative clustering approach to perform robust unsupervised detection of attacks. The main idea is to combine the clustering results provided by multiple independent partitions of the same set of flows, filtering-out biased groupings. The combination of multiple evidence about inter-flows organization adds robustness to the process of separating malicious from normal-operation traffic. The approach combines the notions of Sub-Space Clustering (SSC) [15] and Evidence Accumulation (EA) [16] to produce these multiple independent partitions and to combine the information provided by each of them. Besides detecting network attacks, we show how to use the information provided by the multi-clustering approach to characterize an identified group of malicious flows, automatically producing an easy-to-interpret signature of the attack. This signature provides useful information about the nature of the attack to the network operator, and can be eventually used to expand the list of known-attacks of a signature-based detection system, simplifying its detection in the future.

As a proof-of-concept of how such a robust unsupervised detection approach may work in the practice, we develop a complete system to detect and characterize standard network attacks without any previous knowledge about their existence, testing its performance in real traffic captured in two operational networks: the backbone network of the Japanese WIDE project [22], and the French RENATER research network. In addition, we show that this method outperforms previously proposed methods for unsupervised detection of attacks.

II. Network Attacks

Although we claim that our approach can be used to detect and characterize unknown malicious flows, we focus on the detection and characterization of standard and well-known attacks, which facilitates the interpretation of results. However, we shall assume no previous knowledge about these attacks, and thus treat them as completely unknown. Denial of Service (DoS), Distributed DoS (DDoS), network scans, and worms propagation are examples of standard attacks that daily threaten the integrity and normal operation of the network.

DoS/DDoS: a DoS/DDoS attack [19] is an attempt to make a network resource (a particular service, network bandwidth, etc.) unavailable to its intended (legitimate) users. In its most general form, a DoS/DDoS attack seizes resources by using or requesting more than the victim can handle, preventing it from responding to legitimate requests. A common DoS/DDoS attack is known as SYN flooding, in which the attacker sends a large number of TCP/SYN packets asking for a connection initiation to the victim’s service, which has to keep track of these partially opened connections and can not respond to legitimate requests. Other flooding attacks are based on sending an overwhelming number of ICMP packets to the victim (smurf attack, ping flood, etc.), causing a severe bandwidth exhaustion. DoS attacks are characterized by a single host sending traffic towards a single victim, whereas DDoS involve traffic from many sources towards the same victim.

Worms propagation: a worm [20] is a malicious self-replicating program that uses the network to send copies of itself, infecting other machines by exploiting specific vulnerabilities. A worm is normally used to install a back-door in the infected computer, allowing the creation of a “zombie” machine under the control of the attacker. Networks of such machines are referred to as “botnets”, and are generally used to launch massive DDoS attacks. A worm first scans the network in search of possible victims to infect. During the propagation phase, an infected machine sends traffic to a large number of destinations.

Network scan: a network scan [21] is a probing attempt to identify the availability of a specific service on many different machines. Detecting network scans is extremely important because such an activity is usually a precursor of the propagation of a worm, and therefore the precursor of possible DDoS attacks. Network scans are characterized by a single source sending traffic to many destinations.

III. Unsupervised Detection & Characterization of Network Attacks

The detection algorithm that we present runs in a time sliding-window basis, capturing packets in consecutive time slots of fixed length ΔT. The analysis is performed in three consecutive stages. In the first stage, we use any traditional time-series abrupt-change detection algorithm to detect the presence of an anomalous slot. Modeling anomalies as abrupt-changes in network data time-series is a standard approach [4]–[7]. Packets are aggregated into flows at the end of each slot, and time-series for change detection are built on them, using simple traffic metrics such as number of bytes, packets, or flows per time slot. A flow of packets is defined for source or destination, using either $(\mathrm{IPsrc}/\mathrm{netmask})$ or $(\mathrm{IPdst}/\mathrm{netmask})$ as flow identifier. The use of different netmasks (i.e. $/16$, $/24$, $/32$) provides different levels of traffic aggregation, which facilitates detection in the event of both single source-destination and distributed attacks.

A. Sub-Space Clustering & Evidence Accumulation

The unsupervised detection and characterization algorithm begins in the second stage, using as input the set of flows captured in the anomalous slot. An anomaly is generally detected in different aggregation levels, and there are many heuristics to select a particular aggregation to use in the unsupervised stage; for the sake of simplicity we shall skip this issue, and use any of the aggregation levels in which the anomaly was detected. Without loss of generality, let $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be the set of n flows in the flagged slot. Each flow $\mathbf{y}_i \in \mathbf{Y}$ is described by a set of m traffic attributes or *features*. Table I presents the different features used in this article. The list includes standard and very basic traffic attributes, which permits to describe the detected attacks in easy-to-interpret terms. As we show in the evaluation, such features are good enough to detect and characterize standard network attacks such as DoS, DDoS, and network scans. However, the list is by no means exhaustive, and more features can be easily plugged-in to improve detection and characterization results. Let $\mathbf{x}_i = (x_i(1), \ldots, x_i(m)) \in \mathbb{R}^m$ be the corresponding vector of m traffic features describing flow $\mathbf{y}_i$, and $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ the complete matrix of features, refereed to as the *feature space*.

Feature	Description
nSrcs	n^{o} of sources
nDsts	n^{o} of destinations
nSrcs/nDsts	ratio of nSrcs to nDsts
nSrcPorts	n^{o} of different source ports
nDstPorts	n^{o} of different destination ports
nPkts/sec	n^{o} of packets per second
nPkts/nDst	n^{o} of packets per destination
nICMP/nPkts	fraction of ICMP packets
nSYN/nPkts	fraction of SYN packets

Table I

EXAMPLES OF THE FEATURES USED TO DETECT AND CHARACTERIZE DOS, DDOS, AND NETWORK SCANS.

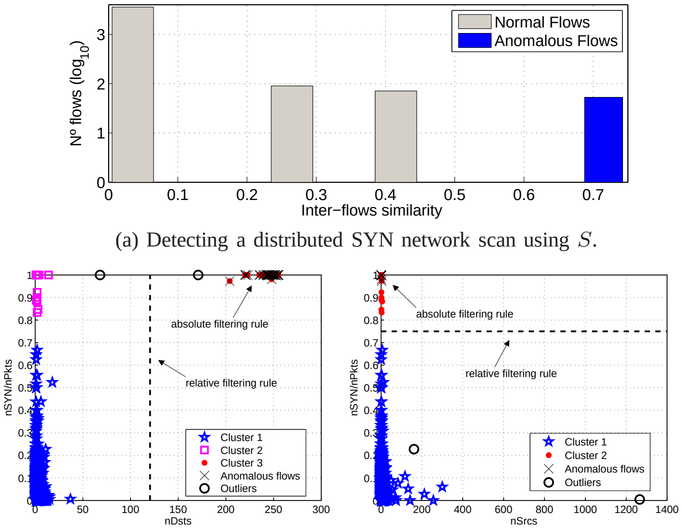
The algorithm is based on clustering techniques applied to $\mathbf{X}$. Our goal is to identify in $\mathbf{Y}$ the different flows that may compose the attack. For doing so, we recall that a large fraction of network attacks are contained in a small fraction of traffic flows. Thus, an attack may consist of either outliers (i.e., single isolated flows) or small-size clusters, depending on the aggregation level of flows in $\mathbf{Y}$. Let us take as an example a DDoS attack launched from β sources distributed along δ different $/24$ botnets towards a single victim. The attack is represented as a cluster of β flows if the aggregation is done for $\mathrm{IPsrc}/32$, or as an outlier if the aggregation is done for $\mathrm{IPdst}/32$. Taking into account that the number of flows in $\mathbf{Y}$ can reach some thousands even for short time slots, the number of sources β would have to be extremely large to violate the assumption of small-size cluster. Besides, if this would be the case, then the attack would be represented as a small-size cluster of $\delta << \beta$ flows when using $\mathrm{IPsrc}/24$ aggregation. In addition, distributed attacks with thousands of sources are easily detected with standard techniques [19] and thus they are less interesting to us, because the evidence of such attacks is overwhelming.

To avoid the lack of robustness of general clustering techniques, we have developed a divide & conquer clustering approach, combining the notions of Sub-Space Clustering and Evidence Accumulation. Instead of directly partitioning the complete feature space $\mathbf{X}$, the SSC-EA-based algorithm does clustering in N different sub-spaces $\mathbf{X}_i \subset \mathbf{X}$ of smaller dimensions, obtaining N different partitions P_i of the flows in $\mathbf{Y}$. Each partition P_i is obtained by applying DBSCAN [18] to sub-space $\mathbf{X}_i$. DBSCAN is a powerful density-based clustering algorithm that discovers clusters of arbitrary shapes and sizes [17]. Each sub-space $\mathbf{X}_i$ is constructed using only $r < m$ traffic features; this permits to analyze the structure of $\mathbf{X}$ from N different perspectives, using a finer-grained resolution. In particular, we do clustering in very-low dimensional sub-spaces, using $r = 2$. To deeply explore the complete feature space, we analyze all the r-combinations-obtained-from-m sub-spaces; hence, $N = m(m-1)/2$. The information provided by the multiple partitions P_i is then combined to produce a new similarity measure between flows in $\mathbf{Y}$, which has the paramount advantage of clearly highlighting both the outliers and small-size clusters that were simultaneously identified in different sub-spaces. This new similarity measure is finally used to easily extract the anomalous flows from the rest of the traffic. Briefly speaking, if we can find single flows or a small group of flows that are remarkably different from the rest of the traffic in different sub-spaces, then we have found an anomaly; if not, the flagged slot was just a false alarm. The simultaneous use of SSC and EA adds robustness to the clustering process, improving the ability of the algorithm to properly detect attacks.

B. Automatic Characterization of Attacks

At this stage, the unsupervised algorithm has identified a set of very similar flows in $\mathbf{Y}$ distant from the majority of traffic. The following task is to automatically produce a set of K filtering rules $f_k(\mathbf{Y})$, $k = 1, \ldots, K$ to correctly isolate and characterize these flows. In the one hand, such filtering rules provide useful insights on the nature of the anomaly, easing the analysis task of the network operator. On the other hand, different rules can be combined to construct a signature of the anomaly, which can be used to detect its occurrence in the future, using a traditional signature-based detection system.

In order to produce filtering rules $f_k(\mathbf{Y})$, the algorithm selects those sub-spaces $\mathbf{X}_i$ where the separation between the anomalous flows and the rest of the traffic is the biggest. We define two different classes of filtering rule: *absolute* rules $f_A(\mathbf{Y})$ and *relative* rules $f_R(\mathbf{Y})$. Absolute rules are



(a) Detecting a distributed SYN network scan using S.

(b) SYN network scan (1/2)

(c) SYN network scan (2/2)

Figure 1. Filtering rules for characterization of a SYN network scan.

only used in the characterization of small-size clusters. These rules do not depend on the separation between clusters, and correspond to the presence of dominant features in the anomalous flows. An absolute rule for feature j has the form $f_A(\mathbf{Y}) = \{\mathbf{y}_i \in \mathbf{Y} : x_i(j) == \lambda\}$. For example, in the case of an ICMP flooding attack, the vast majority of the associated flows use only ICMP packets, hence the absolute filtering rule $\{\text{nICMP}/\text{nPkts} == 1\}$ verifies for them.

Relative filtering rules depend on the separation between anomalous and normal-operation flows. Basically, if the anomalous flows are well separated from the rest of the clusters in a certain partition P_i, then the features of the corresponding sub-space $\mathbf{X}_i$ are good candidates to define a relative rule. A relative rule defined for feature j has the form $f_R(\mathbf{Y}) = \{\mathbf{y}_i \in \mathbf{Y} : x_i(j) < \lambda$ or $x_i(j) > \lambda\}$.

We shall also define a *covering relation* between filtering rules: we say that rule f_1 *covers* rule $f_2 \leftrightarrow f_2(\mathbf{Y}) \subset f_1(\mathbf{Y})$. If two or more rules overlap (i.e., they are associated to the same feature), the algorithm keeps the one that covers the rest.

To construct a compact signature of the anomaly, we shall select the most discriminant filtering rules. Absolute rules are important, because they define inherent characteristics of the anomaly. As regards relatives rules, their relevance is directly tied to the degree of separation between flows. In the case of outliers, we select the K features for which the normalized distance to the normal-operation traffic (represented by the biggest cluster in each sub-space) is among the top-K biggest distances. In the case of small-size clusters, we rank the degree of separation to the rest of the clusters using the well-known Fisher Score (FS), and select the top-K ranked rules. The FS measures the separation between clusters, relative to the total variance within each cluster. To finally construct the signature, the absolute rules and the top-K relative rules are combined into a single inclusive predicate, using the covering relation in case of overlapping.

IV. Evaluation and Discussion

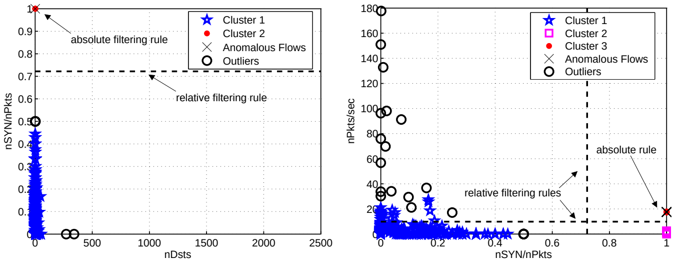
We evaluate the ability of the unsupervised algorithm to detect and to construct a signature for different attacks in real traffic traces from the public traffic repository of the WIDE project [22]. The WIDE operational network provides interconnection between different research institutions in Japan, as well as connection to different ISPs and universities in the U.S.. The traffic repository consists of 15 minutes-long raw packet traces collected since 1999. Traces are not labeled, thus our analysis is limited to show the detection and characterization of different network attacks found by manual inspection in randomly selected traces, such as ICMP DoSs, SYN network scans, and SYN DDoS. In all cases, we shall assume no previous knowledge about these attacks, and thus treat them as completely unknown.

We also test the true positive and false positive rates obtained in the detection of annotated attacks, using different traffic traces from the METROSEC project [23]. These traces consist of real traffic collected on the French RENATER network, containing simulated attacks performed with well-known DDoS attack tools. DDoS attacks range from very low intensity (i.e., less than 4% of the overall traffic volume) to massive attacks (i.e., more than 80% of the overall traffic volume). Additionally, we compare the performance of the algorithm against some previous methods for unsupervised outliers detection based on clustering [9]–[12], as well as against the very well-known Principal Components Analysis (PCA) approach [13]. PCA is a standard technique for unsupervised data analysis, based on dimensionality reduction.

A. Detecting a network scan

We first detect and characterize a distributed SYN network scan directed to many victim hosts under the same $/16$ destination network. Packets in $\mathbf{Y}$ are aggregated in $\text{IPdst}/24$ flows, thus we shall detect the attack as a small-size cluster. The length of each slot is $\Delta\text{T} = 20$ seconds. As we explained in section III-A, the SSC-EA-based clustering algorithm constructs a new similarity measure between flows in $\mathbf{Y}$. We shall express this new similarity measure as a $n \times n$ matrix S, in which element $S(i,j)$ represents the degree of similarity between flows i and j. Figure 1.(a) depicts a histogram on the values of S. The structure of flows provided by S evidences the presence of a small isolated cluster in multiple sub-spaces. Selecting the most similar flows w.r.t. S results in a compact cluster of 53 flows; a further analysis of these flows reveals different $\text{IPdst}/32$ sub-flows of SYN packets with the same IPsrc address, corresponding to the scanning machine.

As regards filtering rules and the associated signature of the attack, figures 1.(b,c) depict some of the partitions P_i where both absolute and top-K relative rules were produced. These rules relate the number of sources and destinations, and the fraction of SYN packets. Combining them produces a signature that can be expressed as $(\text{nSrcs} == 1) \wedge (\text{nDsts} > \lambda_1) \wedge (\text{nSYN}/\text{nPkts} > \lambda_2)$, where λ_1 and λ_2 are two thresholds obtained by separating clusters at half distance. The signature makes perfect sense,



(a) SYN DDoS (1/2)

(b) SYN DDoS (2/2)

Figure 2. Filtering rules for characterization of a SYN DDoS attack.

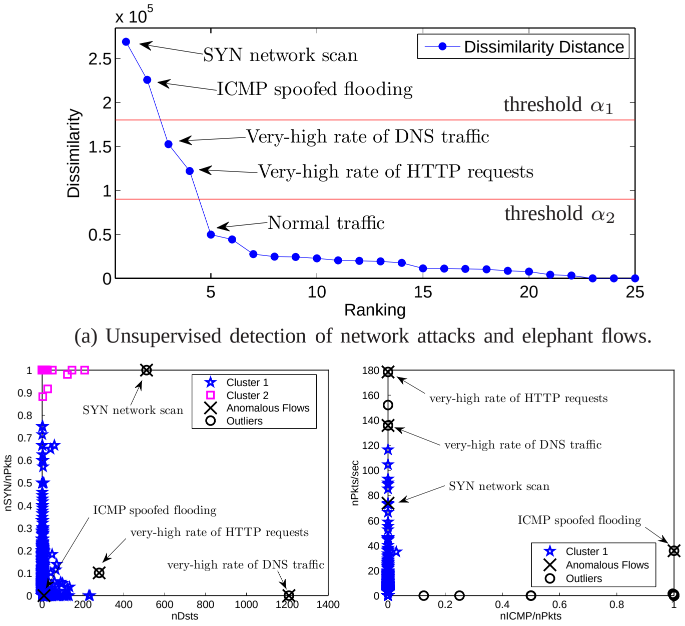
since the network scan sends SYN packets from a single host to a large number of victims. The paramount advantage of the approach relies on the fact that this new signature has been produced without any previous knowledge about the attack or the baseline traffic.

B. Detecting a DDoS attack

Figures 2.(a,b) depict different rules obtained in the detection of a SYN DDoS attack. Traffic is now aggregated in $\text{IPsrc}/32$ flows, and the attack is detected as a small-size cluster. The analysis of inter-flows similarity w.r.t. S selects a compact isolated cluster, corresponding to the set of attacking hosts. The obtained signature can be expressed as $(\text{nDsts} == 1) \wedge (\text{nSYN}/\text{nPkts} > \lambda_3) \wedge (\text{nPkts}/\text{sec} > \lambda_4)$, which combined with the large number of identified sources $(\text{nSrcs} > \lambda_5)$ confirms the nature of a SYN DDoS attack. This signature is able to correctly isolate the most aggressive hosts of the DDoS attack, i.e., those with highest packet rate.

C. Detecting attacks as outliers

In the case of outliers detection, the similarity measure provided by the SSC-EA-based algorithm does not represent inter-flows similarity; instead, it corresponds to the cumulative separation of an outlier to the biggest cluster in the different sub-spaces. The biggest cluster in each sub-space statistically represents normal-operation traffic. Let us first present the detection of a SYN network scan and an ICMP flooding attack using the SSC-EA-based outliers detection approach. Traffic is aggregated in $\text{IPsrc}/32$ flows. Figure 3.(a) shows the ordered dissimilarity values obtained for the different flows, along with their corresponding classification. The first two most distant flows correspond to a highly distributed SYN network scan (more than 500 destination hosts) and an ICMP spoofed flooding attack directed to a small number of victims (ICMP redirect traffic, directed towards port 0). The following two flows correspond to unusual large rates of DNS traffic and HTTP requests; from there on, flows correspond to normal-operation traffic. Note that both attacks can be easily detected and isolated from the anomalous but yet legitimate traffic without false alarms, using for example the threshold α_1. Figures 3.(b,c) depict the corresponding four flows in two of the partitions produced by the SSC-EA-based method. Besides showing typical characteristics of these attacks, both partitions show that the attacks do not represent the largest elephant flows in the time slot. This emphasizes the ability of the algorithm to detect low volume attacks, even of lower intensity than normal traffic.



(a) Unsupervised detection of network attacks and elephant flows.

(b) Scan and flooding attacks (1/2)

(c) Scan and flooding attacks (2/2)

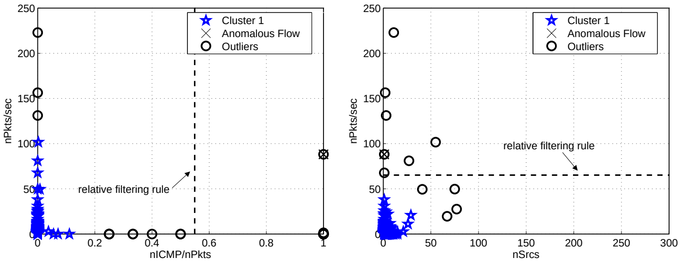
Figure 3. Detection and analysis of network attacks in WIDE.

To conclude, figures 4.(a,b) present the detection and automatic characterization of an ICMP flooding DoS attack. Traffic is aggregated according to $\text{IPdst}/32$. Absolute rules are not applicable in the case of outliers detection. Relative rules correspond to the separation of the outlier from the biggest cluster in each sub-space. Besides showing typical characteristics of this attack, such as a high packet rate of exclusively ICMP packets from the same source, both partitions evidence once again the ability of the algorithm to detect network attacks that are not necessarily the biggest elephant flows. The obtained signature can be expressed as $(\text{nICMP}/\text{nPkts} > \lambda_6) \wedge (\text{nPkts}/\text{sec} > \lambda_7)$.

D. Detecting Attacks with Ground Truth: METROSEC traffic

Figure 5 depicts the True Positives Rate (TPR) as a function of the False Positives Rates (FTR) in the detection of 9 DDoS attacks in the METROSEC dataset. From these 9 attacks, 5 correspond to massive attacks (more than 70% of the traffic), 1 to a high intensity attack (about 40%), 2 are low intensity attacks (about 10%), and 1 is a very-low intensity attack (about 4%). The detection is performed with traffic aggregated in$\text{IPdst}/32$ flows. The ROC plot is obtained by comparing the sorted dissimilarity values obtained for the different flows to a variable detection threshold. The SSC-EA-based algorithm can correctly detect 8 out of the 9 attacks without false alarms. The detection of the very-low intensity attack is more difficult; however, the 9 attacks are correctly detected with a very low FPR, about 1.2%.

We compare the performance of our approach against three previous unsupervised approaches: DBSCAN-based, k-means-



(a) ICMP echo DoS (1/2)

(b) ICMP echo DoS (2/2)

Figure 4. Filtering rules for characterization of an ICMP DoS attack.

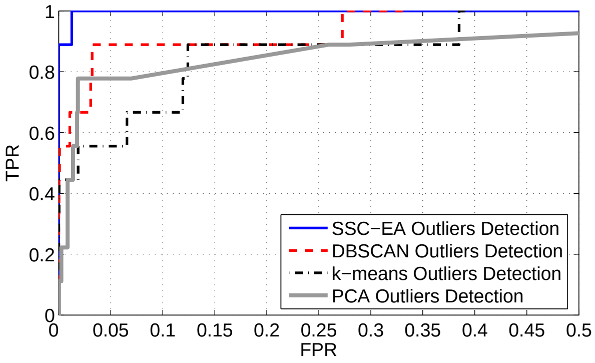


Figure 5. DDoS detection in METROSEC. The SSC-EA-based method is able to detect even a very-low intensity DDoS attack with a very small false alarm rate, which is not possible with traditional unsupervised approaches.

based, and PCA-based outliers detection. The first two consist in applying two standard clustering algorithms, either DBSCAN [18] or k-means [17] to the complete feature space $\mathbf{X}$, identify the largest cluster C^{*}, and compute the distance of all the flows lying outside C^{*} to its centroid. The ROC is finally obtained by comparing the sorted distances to a variable detection threshold. These approaches are similar to those used in previous work [9]–[12]. In the PCA-based approach, PCA and the sub-space methods [13], [14] are applied to the complete matrix of features $\mathbf{X}$, and the attacks are detected by comparing the residuals to a variable threshold. Both the k-means and the PCA-based approaches require fine tuning: in k-means, the number of clusters k to identify must be set a-priori; in PCA, the number of principal components used to describe normal-operation traffic must be decided with heuristics. In the case of k-means, we repeat the clustering for different values of clusters k, and present the average results. In the case of PCA we present the best performance, obtained while using 2 principal components to describe normal-operation traffic.

Obtained results permit to evidence the great advantage of using the SSC-EA-based algorithm in the clustering step w.r.t. to traditional approaches. In particular, all the approaches used in the comparison fail to detect the smallest attack with a reasonable false alarm rate. Both the DBSCAN-based and the k-means-based algorithms get confused by masking features when analyzing the complete feature space $\mathbf{X}$. As evidenced in previous work [14], the PCA approach is not sensitive enough to discriminate both low-intensity and high-intensity attacks, using the same representation for normal-operation traffic.

V. Implementation Issues

The SSC-EA-based algorithm performs clustering in $N = m(m-1)/2$ low-dimensional sub-spaces $\mathbf{X}_i \in \mathbb{R}^2$. As we have shown, this provides a high discrimination power to detect and characterize different types of network attacks. However, the multiple clusterings computation increases the total Computational Time (CT) of the algorithm, imposing scalability issues for on-line detection of network attacks in very-high-speed networks. Scalability should be addressed as regards both the number of features used to describe traffic flows (m) and the number of flows to analyze (n). In the real traffic evaluations that we have presented, the number of flows captured in a time slot of $\Delta\mathrm{T} = 20$ seconds rounds $n = 2500$ flows. For the $m = 9$ features that we have used, the total number of clusterings to compute is $N = 36$, which takes about 14.4 seconds in a standard single-processor machine.

Two key features of the SSC-EA-based algorithm can be exploited to reduce scalability problems in m and n. Firstly, clustering is performed in low-dimensional sub-spaces ($\mathbb{R}^2$), independently of the number of features that are used. Clustering in low-dimensional feature spaces is faster than in high-dimensional spaces [17], which partially alleviates the overhead of multiple clusterings computation. Secondly, the clustering of each sub-space $\mathbf{X}_i$ can be performed independently of the analysis on the other sub-spaces, which is perfectly adapted for parallel computing architectures. Parallel computing has become the dominant paradigm for accelerating specific tasks and represents a booming domain, driven by the availability of strong computational-power entities at low costs. Parallelization can be achieved in different ways: using a single multi-processor and multi-core machine, using GPU (Graphic Processor Unit) capabilities, using network-processor cards, using a distributed group of machines, or combining these techniques. We shall use the term ”slice” as a reference to a single computational entity.

Modern network-processor cards are able to perform traffic monitoring even in 10 Gbps network connections. In a average-loaded 10 Gbps link (about 50%-60%) there are about 500.000 packets per second; if we consider traffic flows with an average rate of 500 kbps (about 50 pkts/sec) and a average duration of at least 20 seconds, then we have about $n = 10.000$ flows to analyze in each time slot of $\Delta\mathrm{T} = 20$ seconds. From our experimentations, we known that we can analyze this number of flows using as much as $m = 20$ traffic descriptors in less than 20 seconds, using a parallel architecture with about 100 slices. Current network-processor cards vendors offer multi-core solutions for high-performance networking with as much as 64 general purpose cores [24], which are perfectly adapted to deploy our unsupervised detection algorithm for very-high-speed knowledge-independent traffic monitoring.

VI. Concluding Remarks

In this article we question the ability and stress the limitations of current knowledge-based approaches for detection of network attacks, particularly in the context of an increasingly complex and ever-evolving Internet. In a diametrically opposite perspective, we place the emphasis on the development of unsupervised, knowledge-independent detection algorithms, which we believe is the next natural step in network traffic monitoring for network security.

As a proof-of-concept of how such a detection approach could be actually implemented in the practice, we have presented a robust multi-clustering-based detection method and evaluated its ability to detect and characterize standard network attacks without any previous knowledge, using packet traces from two real operational networks. In addition, we have shown detection results that outperform previous proposals for unsupervised detection of attacks, providing more evidence of the feasibility of an accurate knowledge-independent detection system.

To conclude, we have briefly discussed implementation issues of the presented approach, showing that its use for on-line unsupervised detection and automatic generation of signatures is a-priori possible, even while using more traffic descriptors to characterize network attacks, and even when running in very-high-speed networks.

Acknowledgments

This work has been done in the framework of the ECODE project, funded by the European commission under grant FP7-ICT-2007-2/223936.

References

[1] S. Hansman, R. Hunt "A Taxonomy of Network and Computer Attacks", in *Computers and Security*, vol. 24 (1), pp. 31-43, 2005.

[2] M. Roesch, "Snort: Lightweight Intrusion Detection for Networks", in *Proc. USENIX 13th Systems Administration Conference*, 1999. Open-source network intrusion prevention and detection system available at http://www.snort.org/.

[3] D. Denning, "An Intrusion Detection Model", in *IEEE Trans. Soft. Eng.*, vol. 13 (2), pp.222-232, 1987.

[4] M. Thottan and J. Chuanyi, "Anomaly Detection in IP Networks", in *IEEE Trans. Sig. Proc.*, vol. 51 (8), pp. 2191-2204, 2003.

[5] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies", in *Proc. ACM IMW*, 2002.

[6] J. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring", in *Proc. 14th Systems Administration Conference*, 2000.

[7] A. Soule, K. Salamatian, and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection", in *Proc. ACM IMC*, 2005.

[8] G. Androulidakis, V. Chatzigiannakis, and S. Papavassiliou, "Network Anomaly Detection and Classification via Opportunistic Sampling", in *IEEE Network*, vol. 23 (1), 2009.

[9] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering", in *Proc. ACM DMSA Workshop*, 2001.

[10] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data", in *Apps. of Data Mining in Comp. Sec.*, Kluwer Publisher, 2002.

[11] K. Leung and C. Leckie, "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clustering", in *Proc. ACSC05*, 2005.

[12] L. Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V. Kumar, P. Dokas "The MINDS - Minnesota Intrusion Detection System", in *Next Generation Data Mining*, MIT Press, 2004.

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

[14] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection", in *Proc. ACM SIGMETRICS*, 2007.

[15] L. Parsons, E. Haque, and H. Liu, "Subspace Clustering for High Dimensional Data: a Review", in *ACM SIGKDD Expl. Newsletter*, vol. 6 (1), pp. 90-105, 2004.

[16] A. Fred and A. K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation", in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27 (6), pp. 835-850, 2005.

[17] A. K. Jain, "Data Clustering: 50 Years Beyond K-Means", in *Pattern Recognition Letters*, vol. 31 (8), pp. 651-666, 2010.

[18] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", in *Proc. ACM SIGKDD*, 1996.

[19] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms", in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34 (2), pp. 39-53, 2004.

[20] C. Zou, W. Gong, D. Towsley, and L. Gao, "The Monitoring and Early Detection of Internet Worms", in *IEEE/ACM Trans. Net.*, vol. 13 (5), pp. 961-974, 2005.

[21] G. Xiaobing, Q. Depei, L. Min, Z. Ran, and X. Bin, "Detection and Protection Against Network Scanning: IEDP", in *Proc. ICCNMC 2001*, 2001.

[22] K. Cho, K. Mitsuya, and A. Kato, "Traffic Data Repository at the WIDE Project", in *USENIX Annual Technical Conference*, 2000.

[23] "METROlogy for SECurity and QoS", at http://laas.fr/METROSEC

[24] "JumpGen Network-Processor Cards", at http://www.jumpgen.com