Operational Experiences with Anomaly Detection in Backbone Networks

Maurizio Molina[a,*], Ignasi Paredes-Oliva[b,**], Wayne Routly[a], Pere Barlet-Ros[b]

[a] *DANTE*
126-130 Hills Road, Cambridge CB2 1PQ, United Kingdom
[b] *UPC BarcelonaTech*
Jordi Girona, 1-3, 08034 Barcelona, Spain

Abstract

Although network security is a crucial aspect for network operators, there are still very few works that have examined the anomalies present in large backbone networks and evaluated the performance of existing anomaly detection solutions in operational environments. The objective of this work is to fill this gap by reporting hands-on experience in the evaluation and deployment of an anomaly detection solution for the GÉANT backbone network. During this process, we analyzed three different commercial tools for anomaly detection and then deployed one of them for several months in the 18 points-of-presence of GÉANT. We first explain the general requirements that an anomaly detection system should satisfy from the point of view of a network operator. Afterwards, we describe the evaluation of the tools and present a study of the anomalies found in a continental backbone network after operationally using the finally deployed tool for half a year. We think that this first hand information can be of great interest to both professionals and researchers working on network security and can also guide future research towards more practical problems faced by network operators.

Keywords: Network Security, Anomaly Detection, Benchmarking, NetFlow, Network Management

1. Introduction

Network operators have always been interested in keeping track of the anomalies happening in their network. Traditionally, they have focused on operational (e.g., link faults), or traffic and routing anomalies, observable via SNMP. More

[*] Present affiliation: Open Systems AG (Räffelstrasse 29, 8045 Zurich, Switzerland)
[**] Corresponding author (Tel: +34 934017182, Fax: +34 934017055)
Email addresses: maurizio.molina@gmail.com (Maurizio Molina), iparedes@ac.upc.edu (Ignasi Paredes-Oliva), wayne.routly@dante.net (Wayne Routly), pbarlet@ac.upc.edu (Pere Barlet-Ros)

recently, there has been a business driver for observing anomalies related to security issues, network abuse, or IPR violation.

The reason for investing in security, even in core networks, is that offering a more secure network (i.e., protecting customers from external or internal threats) is becoming a differentiating factor for ISPs, already offering managed security services to their business customers. Furthermore, commercial peering agreements between ISPs often include commitments to avoid transferring potentially harming traffic (e.g., DoS attacks).

Detecting security anomalies requires a more granular visibility of the network than what can be provided by SNMP traffic counters. NetFlow [1] is becoming one of the primary sources of information for security services.

GÉANT [2] is a multi-Gigabit backbone network interconnecting the European National Research and Education Networks (NRENs). DANTE [3], as the operator of GÉANT, is uniquely positioned to provide added value to the security work of NREN CERTs[1]. For example, Distributed DoS attacks can be mitigated and filtered closer to the source of the attack. Worm spreading patterns can also appear more clearly when observed on the backbone network interconnecting all the European NRENs rather than separately on each of them.

During fall 2008, DANTE analyzed three commercial tools for anomaly detection. One year after (fall 2009), one of those tools was permanently deployed in the GÉANT network. This work reports on the benchmarking of the tools and the results obtained after using the finally deployed software for half a year.

The novelty of this paper is twofold. First, we report on the limitations of current commercial tools and discuss some aspects that still need further research from the perspective of a network operator. Second, we provide a long-term study of the anomalies occurring in a continental backbone network. Although there is already considerable work in the design of anomaly detection methods, information regarding the type and characteristics of network anomalies in operational networks is rather scarce in the literature. To the best of our knowledge, this is the first study that provides this sort of feedback.

After manually analyzing more than 1000 attacks, we found that, surprisingly, the overlap among the anomalies detected by different tools is extremely low. This is a clear indicator that false negatives are still significant even when comparing commercial tools that are supposed to detect the same sort of anomalies. In addition, our study reveals that *Network Scan* attacks are the most persistent and shows that there are certain geographical regions that are predominant when looking at the top attackers or targets respectively.

We believe that the analysis and results provided in this paper are particularly interesting for both practitioners and researchers working on anomaly detection. For professionals (other operators or companies) willing to deploy a similar solution, this work can provide very useful information ranging from

[1] A CERT(Computer Emergency Response Team) is a group of experts that takes care of any security-related event threatening a NREN.

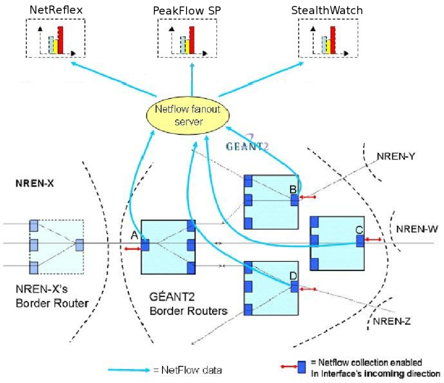


Figure 1: NetFlow collection scenario in the GÉANT network

the requirements used to build the list of anomaly detection tools for the trial, to the evaluation of the tools and the followed methodology. Concerning researchers, the most relevant part of this paper lies on the long-term analysis of anomalies, which can help them in better directing their efforts towards limitations of current commercial products and real threats happening in backbone networks. Additionally, the knowledge on the requirements of a network operator is of great importance and can serve as a guideline to design algorithms able to work in real world environments. Furthermore, we must take into account that having access to commercial solutions is rather uncommon, especially for researchers.

The remainder of this paper is organized as follows. Section 2 describes the requirements used by DANTE to build a short-list of suitable anomaly detection tools. Afterwards, Section 3 reports on the differences found among those tools during the evaluation phase in terms of usability, true and false positives, false negatives and also regarding the different types of anomalies detected. Section 4 presents a study of the network anomalies found in GÉANT along with their properties after using the selected tool for approximately six months. Finally, Section 5 explains in more detail the anomaly detection approaches used by each tool while Section 6 summarizes and concludes the paper.

2. Scenario and Requirements

In this section, we first describe the GÉANT network scenario, where the tools have been analyzed. Then, the requirements used by DANTE to build the short-list of anomaly detection tools are explained and, finally, the selected tools are presented.

2.1. Context and Scenario

DANTE is a non-profit organization that plans, builds and operates the GÉANT backbone network. GÉANT is a /19 transit network connecting 34 European NRENs with 18 points-of-presence (PoPs) spread over Europe (with 10Gb/s links almost everywhere), a dozen of non-european NRENs, and two commercial providers (Telia an Global Crossing). It is the main interconnection point for inter-NREN traffic. For a certain subset of NRENs, GÉANT is also the primary gateway to the commercial Internet (other NRENs have their own connection to the non-research world). Although it is a R&E network, more than half of the traffic is towards commercial providers. The overall handled traffic is more than 50Gb/s.

DANTE collects Sampled NetFlow [4] from every router interface with an external peering network. As GÉANT is a purely transit network, this setup is sufficient to account for all the traffic.

During fall 2008, DANTE started looking for a solution to enhance the security of its network, and of its customer networks, by analyzing three different anomaly detection commercial products. After evaluating the performance of each tool with the same input data for several months, one of them was permanently deployed in the GÉANT backbone network (mid November 2009).

At the beginning of this study, the sampling rate in Sampled NetFlow was set to 1/1000. Later on, the routers were replaced, which allowed to migrate to 1/100 sampling. Therefore, we must take into account that the analysis of the tools presented in Section 3 (fall 2008) and the study presented in Section 4 (2009-2010) were done under different sampling rates (1/1000 and 1/100 respectively).

NetFlow v5 was used since anomaly detection tools require visibility on very granular flows (the ones defined by the 5-tuple src/dst IP, src/dst port and protocol), which is the default (and only one) provided by NetFlow v5. The NetFlow traffic is exported to a single fanout box duplicating it towards multiple destinations (see Figure 1). This setup allowed us to evaluate all the anomaly detection tools using exactly the same input data.

2.2. Tool Requirements

In order to start the process to select one tool for anomaly detection, three candidate tools were short-listed on the basis of a set of requirements. We think that those requirements are representative enough to be useful for any other network operator or company willing to deploy a similar solution. In addition, we think that they impose a serious set of limitations that should be taken into account by researchers working on building anomaly detection algorithms meant to work in real-world networks. Next, we provide the list of requirements along with a brief explanation for each one.

1. Sampled NetFlow support. Given the large scale and traffic volume in backbone networks, one of the main requirements of network operators is the ability of the tools to work with sampled flow-level data (e.g., Sampled NetFlow [4]). Several state-of-the-art tools require access to packet payloads, which renders these solutions impractical for this environment. Recent studies [5–7] have shown that the accuracy of certain anomaly detection techniques is dramatically affected under sampling.

2. Non intrusive collection of data. Tools require often other data, beyond NetFlow. Specifically, sometimes configuration information about the routers needs to be collected to build a tool representation of the network topology and/or to correlate it with information contained in the NetFlow records (e.g., the interface id). Some tools require collection of BGP data, some also IS-IS[2]. The collection of other data should not impose the deployment of additional hardware or difficult configuration changes in the routers. Since most of GÉANT's customers and peering connectivity points are on 10 Gb/s lines, for cost and deployment complexity, approaches requiring the installation of dedicated probes are not appropriate.

3. Accurate detection and classification. For an operator, it is essential to be able to differentiate the anomaly type (correct anomaly classification), to report the end hosts involved and to detect the anomaly duration. It is also very important to detect both the start and the end time of the anomaly (anomaly window) with a precision in the order of several minutes. Based on DANTE's NOC (Network Operations Center) engineers experience, the delay between the true event and its detection should not exceed 20-30 min. and false positives should be low enough in order to be treatable by an operator (e.g., no more than 10-15 anomalies per day).

4. Collection of evidence related to anomalies. Collecting information about the anomalies in a structured way is important to investigate and possibly mitigate the anomalies in collaboration with other CERTs. Relevant information includes: IP addresses and ports, time of the incident and entry/exit network points (both routers and peers).

5. Scalability. The scale of the GÉANT network and the type of traffic posed a significant requirement. At the time of this study, GÉANT had around 10 million unique speaking hosts per day on network with global connectivity, mainly composed of 10 Gb/s links carrying a mixture of research (e.g., grid traffic) and more "ordinary" Internet traffic. Thus, the problem was to detect anomalies with a huge number of IPs and large volumes of composite traffic.

Another important requirement taken into account in DANTE's case was to have tool support. For this reason, only commercial solutions were considered.

[2] BGP (Border Gateway Protocol) is the most commonly used protocol to exchange routing information between Autonomous Systems while IS-IS (Intermediate System To Intermediate System) is limited to an administrative domain or network.

2.3. Analyzed Tools

As a result of the above requirements, three commercial tools were shortlisted: *NetReflex* [8], *PeakFlow SP* [9] and *StealthWatch* [10]. These three tools represent a good cross-section of current best practice techniques in anomaly detection. Moreover, they are all based on different approaches and, therefore, it will be possible to catch a potentially broader range of anomalies with the same input.

Concerning their working scheme, although we cannot go deep into the internal details about what exact algorithms they use due to the fact that they are proprietary, we give below an overview of their main functionalities and architectures. For further details about each particular anomaly detection method these tools are based on refer to Section 5.

2.3.1. NetReflex (NR)

Functionality Overview

NetReflex is a non-intrusive system providing real-time and network-wide visibility. By collecting and processing traffic and routing information, *NR* is capable of performing the following three tasks: topology analysis, traffic analysis and anomaly analysis. The first task focuses mainly on auto-discovering the topology of the network. The traffic analysis task performs real-time inspection of the traffic and, finally, the last task focuses on detecting and classifying anomalies.

Architecture

NR consists of a single physical appliance that integrates all functionalities described above. The system is splitted into the following five core parts: topology analysis, traffic analysis, anomaly analysis, search engine and reporting system. In the first component the system provides information on the topology of the network and other information such as the flows entering or exiting a single PoP or the utilization of a particular link. The traffic analysis functionality computes the traffic matrix. With this information an operator can easily spot the PoP pairs exchanging most of the traffic. In the anomaly analysis component, it reports the anomalies detected along with their type (e.g. DDoS) and all the related meta-data (entering and exiting PoP, source or destination IP, destination port, etc.). The system also provides a search engine, that gives access to the raw NetFlow data and allows the user to perform queries based on the IPs, ports, protocol, entering PoP, etc. Finally, the reporting component provides several types of summaries such as anomaly reports or traffic activity at different levels (e.g., PoP-to-PoP or AS-to-AS).

Anomaly Detection Approach

It uses a technique based on a recent research work [11, 12] that employs Principal Component Analysis (PCA). It applies both volume and entropy metrics along with PCA to discriminate what is normal and what is not. It fuses NetFlow, BGP and IS-IS data and creates a PoP to PoP matrix (18x18 in the GÉANT's case). The PoP to PoP traffic is the elementary unit over which the detection algorithm is run, so that every detected anomaly can be attributed to one uni-directional PoP pair. The fusion of different sources of data and algorithms has several advantages. First, the use of routing data can split the traffic into PoP-PoP pairs and enable the anomaly detection on a level of granularity that is useful for taking corrective actions. Second, PCA allows the automatic compensation of the higher variability of the traffic that some PoP-PoP pairs may "naturally" have. Finally, entropy-based metrics enable the detection of low volume anomalies that cannot be detected using only metrics based on the variation of volume.

2.3.2. PeakFlow SP (PF)

Functionality Overview

PeakFlow is a network-wide system that correlates flow data, SNMP and routing information to build logical models and learn what network behaviours are normal. The feedback provided by these models is then used by operations staff to detect and mitigate anomalies, improve network performance and make better decisions for traffic management and capacity planning. The main difference between *PF* and the other two products presented in this work is that this tool is the only one providing protection besides detection. It is able to keep crucial services such as the DNS/web servers running after detecting a threat towards them.

Architecture

It consists of five types of appliances: the Collector Platform (CP), the Flow Sensor (FS), the Business Intelligence (BI), the Portal Interface (PI) and the Threat Management System (TMS). The CP is placed in the backbone or in a peering edge and takes care of collecting the NetFlow data. The FS, which is placed in the client edge, extends network security to the customer. The BI appliance analyzes the network and reports on its performance (e.g., applications being used). The PI component gives access to the service by providing an interface. It can have multiple instances. For example, in the case of GÉANT, each customer (i.e., NREN) could have its own user interface. Finally, the component taking care of security is the TMS. This part of the software is in charge of detecting the anomalies and applying the proper countermeasures to block them while allowing the flow of legitimate traffic.

Anomaly Detection Approach

This software uses statistical-based and signature-based anomaly detection. Regarding the statistical analysis, it detects anomalies on the basis of variation of traffic volumes. It first creates baseline definitions and then compares the real-time traffic against it to look for abnormal deviations. As for the signatures, it tries to match previously stored patterns with the incoming traffic. Although the statistical base of the anomaly detection of this tool is one of the oldest in the market, the tool has the potential benefit of being easily configurable and using a common Knowledge Base leveraging a quite large installation base. Moreover, several customers (around 50 at the time of the test) provide voluntarily their anomaly feeds to the vendor, who has thus the ability to create new signatures triggering anomalies.

2.3.3. StealthWatch (SW)

Functionality Overview

The *StealthWatch* system provides insight about what applications and services are running in the network, how are they performing, and who is using them. Moreover, it uses behavioural-based analysis to detect security anomalies. Taking into account all this information, it allows IT teams to have more detailed insight and make more reliable decisions for crucial tasks such as incident response, troubleshooting or capacity planning.

Architecture

It is divided into five components: Management Console (MC), Flow Collector (FC), Flow Replicator (FR), Flow Sensor (FS) and Identity (ID). The MC is the user interface through which an operator is able to see graphical representations of what is going on in the network (in terms of both security and usage). The FC takes care of collecting the NetFlow data. The FR component is able to aggregate multiple data sources (e.g., NetFlow, SNMP) in a single data stream and forward it to one or more destinations. The FS is in charge of identifying those applications being used across the network. Finally, the ID part maps any unexpected network event with the user or group of users who caused it.

Anomaly Detection Approach

This system employs behavioural-based analysis. Traffic sent or received by hosts is observed for a number of days (learning phase) and then, the host is classified into the best fitting category according to this profiling (e.g., end host or web server). Deviations from what is believed the “normal” behaviour of the host lead to the triggering of anomalies. This tool requires the manual feed of static BGP prefixes to cluster the IPs in groups. Despite the potential scalability weakness of per-host profiling (see Section 3.6.2), it can be very accurate and precise for detecting sudden anomalous behaviour of single hosts (often related to suspicious of malicious activity). The statistical analysis done in the background is complex. However, the user of this tool has the possibility to easily vary the sensitivity of a host or group of hosts to reduce the number of false positives or to whitelist non-interesting anomalies for the specific scenario where the tool is being used.

3. Analysis of the Tools

In this section, we analyze the three tools presented in Section 2. Firstly, the dataset and the methodology followed during the analysis are explained. Afterwards, an analysis of the true and false positives, the false negatives, the type, and distribution of the anomalies is provided for each tool. Finally, we

Table 1: Details for the datasets

Label	Duration	Period	#flows/#packets/#bytes	Sampling
dataset-1	13 days	Nov.'08	1.97G/4.12G/3.21T	1/1000
dataset-2	175 days	Nov.'09-May'10	99.38G/699.95G/576.16T	1/100

observe how the origins of the anomalies are split in order to see if any of the tools has any bias in the detection.

3.1. Dataset and Methodology

The analysis of the tools is based on a 13 days long dataset collected during November 2008 (days 9-12, 16, 18-22, 23-26) with a sampling rate of 1/1000. We refer to this dataset as *dataset-1* (see details in Table 1).

Every single anomaly inside *dataset-1* was manually analyzed via access to the raw NetFlow records by a DANTE security team with long experience in network security. Some of the anomalies (especially unclear cases) were double-checked with NRENs to obtain an independent validation. An overall of 1006 anomalies were manually analyzed.

Each anomaly was classified either as a true positive (TP), a false positive (FP) or as an Unknown (U). TP means that there was enough evidence to confirm that the event was indeed due to a malicious activity, whereas a FP implies that there was a clear indication that the detected anomaly corresponded to legitimate traffic. An anomaly was classified as an unknown when the security team was not able to reach a conclusion and, therefore, they were not able to confirm if it was just normal traffic (FP) or an actual anomaly (TP). A small sample of false negatives (FN) was also analyzed as discussed in Section 3.4.

3.2. Type of Anomalies

We analyzed the type and distribution of the anomalies reported by each tool. Since not all tools signaled exactly the same sort of attacks, we classified all the reported alarms into the security categories below.

Network Scans (*NS* or horizontal scans) are carried out by sending probes to identify running services on a network. Those probes are always sent to one specific port (or a few), but to a multitude of destinations.

Port Scans (*PS* or vertical scans) are aimed at detecting running services on a specific machine. Essentially, this type of scan consists of sending a message to a huge amount of ports, one at a time. The kind of response received indicates whether the port is used and can therefore be further probed for weakness.

Denial-of-Service (*DoS*) is an attack on a computer system that floods the network or the end system. They are attempts to make a resource unavailable to its users. Most common *DoS* categories we found during this work were UDP floods and TCP SYN floods.

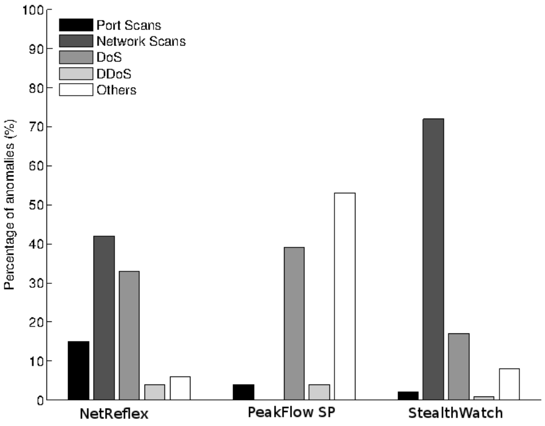


Figure 2: Anomalies reported by each tool

Distributed DoS (*DDoS*) attacks can take the forms described for DoS, but the senders are a multitude of (often compromised) systems attacking a single target. The effects of distributed attacks are nastier and their mitigation more difficult.

Others category was used to separate events that did not fit in the groups above.

As we can observe in Figure 2, *NetReflex (NR)* and *StealthWatch (SW)* detected events falling into all categories, while *PeakFlow SP (PF)* missed completely all the *Network Scans*. This sort of attack is clearly the most frequent one according to *NR* and *SW* (42% and 72% respectively). Even though with different percentages, these two tools also coincided classifying the second more frequent attack, the *DoS* (33% and 17%), and the least common, the *DDoS* (4% and < 1%). The most significant discrepancy left was the amount of reported *PS*: while *NR* detected quite a lot of them (14%), *SW* only triggered 2%.

PF presented quite different results showing a proportion of 39% of *DoS* attacks and 4% for both *Port Scans* and *DDoS* (respectively). In addition, according to this tool, more than half of the detected anomalies (53%, almost all of them FP, as explained in Section 3.4) belonged to the *Others* category while *SW* and *NR* showed significantly smaller percentages for that group (8% and 6% respectively). Note that all tools were working under an aggressive sampling rate during the evaluation period (1/1000) and therefore this could have a significant impact on their accuracy.

The results clearly reflect the strong points of the methods the tools are based on. For example, *SW*, based on per-host behavioural analysis, was the strongest detecting *Network Scans* because when there is scanning activity, the behaviour of a host changes significantly. *PF*, that uses a baseline to detect abnormal volume variations, was the one detecting more *DoS* because this sort

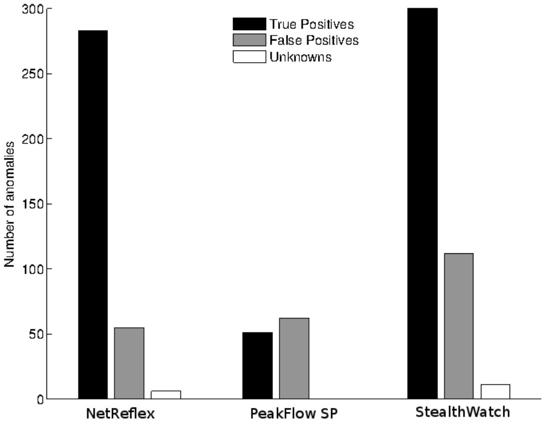


Figure 3: True Positives, False Positives and Unknowns for the evaluated tools

of attack uses large amounts of packets or bytes. Finally, *NR*, based on entropy, showed the best balance among the four types of anomalies.

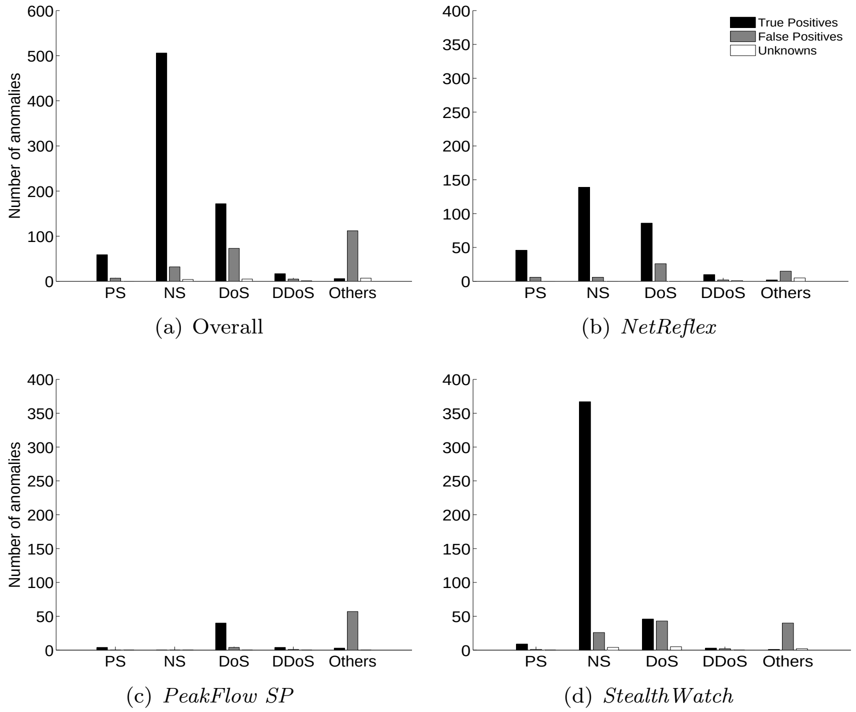
3.3. True and False Positives Analysis

Concerning the figures per tool (see Figure 3), *SW* was the one detecting more anomalies (549) followed by *NR* (344). *PF* was the one with the smallest set of reported anomalies: 113. Regarding the True Positives (TP), *NR* had the best ratio (82.26%) followed closely by *SW* (77.59%), while *PF* showed the worst performance with 45.13%. The false positive (FP) ratio for *NR* was the lowest one (15.98%), while *SW* had a similar value (20.4%). Half of the anomalies detected by *PF* (54.86%) were FP. Therefore, *NR* clearly showed the best ratio TP-FP, although it detected far less anomalies than *SW*. Regarding the "Unknowns" category, *NR* and *SW* had just few cases (≈2% each) and *PF* did not have any.

Figure 4 shows the total number of TP, FP and Unknowns per anomaly type for each tool and also taking into account all tools together. As we can clearly observe in Figure 4(a), the amount of overall FP compared to the number of TP was reasonably low for *PS*, *NS* and *DDoS*. On the contrary, the false positive ratio was non negligible in the case of *DoS* (29.2%), while the *Others* category was almost purely composed by FP. Figure 4(c) shows that the FP in the former group were basically signaled by *PF*. The false positives in case of *DoS* were mainly because of *NR* and *SW*. For *NR* (Figure 4(b)), they were almost one-third of the TP and, for *SW* (Figure 4(d)), the FP were, strangely, as big as the number of true positives.

3.4. Anomaly Overlap and False Negatives Analysis

For an anomaly detection system, the tradeoff between false negatives (FN) and FP is very important. Since we knew the set of true anomalies for all tools and also their intersection, we were able to compute a lower bound of the FN



(a) Overall

(b) *NetReflex*

(c) *PeakFlow SP*

(d) *StealthWatch*

Figure 4: True Positives, False Positives and Unknowns per anomaly type

for each tool. For a given tool, we know that its lower bound of FN is composed by all TP flagged by the other tools but not detected by the tool itself.

However, analyzing the full set of FN was extremely hard in our environment, because it requires manual inspection of every single flow (i.e., 1.97×10^9 flows, not only those belonging to anomalies detected by the tools) to determine whether it is part of an anomaly. We discarded other alternatives, such as performing a penetration test, due to legal issues since the analyzed sources and destinations were outside the administrative domain of DANTE.

Surprisingly, the intersection among the set of anomalies detected by all tools was limited to a few percent. *NR* had only 17 anomalies in common with *PF* and 29 with *SW*. *PF* and *SW* only shared 6 anomalies. This is a strong indication that the particular anomaly detection approach used by each tool clearly influences what kinds of anomalies are reported, even though all tools aim to the detect the same type of events. This highlights the importance of combining different anomaly detection systems to catch a broader range of anomalies.

Concerning the lower bound of false negatives per anomaly type, they all presented huge values as we can observe in Table 2 (last row). They were approximately twice as big as the TP for both *NS* and *DoS* (respectively). For

Table 2: Lower bound of false negatives per tool and anomaly type for *dataset-1*

	PS	NS	DoS	DDoS	Others	Overall
NetReflex	9	350	67	3	1	430
PeakFlow SP	50	505	122	3	2	682
StealthWatch	40	123	113	12	4	292
Overall	99	978	302	18	7	1404

DDoS, they were almost equal to the TP whereas *PS* showed the lowest ratio (close to two-thirds of the TP). Regarding the overall number of FN for each tool, *PF* was the one with the highest value (682) followed by *NR* (430) and *SW* (292). Note that although all tools detected approximately the same sort of anomalies, the lower bound of false negatives for the tool with the lowest value (*SW*), already indicates that there were at least 67.91% more anomalies happening in the network besides those being detected by the tool itself.

In order to confirm that the false negatives were so significant, we performed an alternative analysis based on a subset of all the FN. The newly created ground truth of anomalies was manually validated and created independently of the tools. We used *frequent itemset mining* (FIM), a data mining technique that has been recently used in the literature to extract sets of anomalous flows [13, 14]. We randomly selected sixteen 30-minute samples of NetFlow within *dataset-1* and run FIM on them. Afterwards, we manually splitted the reported sets of flows into legitimate and anomalous traffic. In case of being anomalous, we also classified them taking into account the types of anomalies described in Section 3.2. This final set of anomalies with their corresponding type was our ground truth. Therefore, any anomaly in this ground truth that was not detected by a particular tool was considered a false negative for that specific tool. FIM has an input parameter called *minimum support (ms)* that determines how big the set of flows must be in order to be reported. In our case, the *ms* is specified in terms of flows. We used *ms=2000*, which resulted in a reasonable number of anomalies to treat manually. Accordingly, this analysis of false negatives was limited to those anomalies reported by *NR*, *PF* or SW that had 2000 flows or more. For the analyzed intervals of time and having at least 2000 flows, *NR*, *PF* and *SW* had 8, 5 and 4 anomalies respectively.

Although all the anomalies flagged by *NR*, *PF* and *SW* were found using FIM for the analyzed periods, we found many more anomalies that had not been detected by any of the three tools. In particular, the manually validated ground truth had 126 anomalies. Therefore, the lowest percentage of FN was for *NR* (93.65%), closely followed by *PF* (96.03%) and *SW* (96.83%). This new evaluation further certified that even current commercial tools are still missing a vast amount of anomalies.

3.5. Origin of the Anomalies

As described in Section 2.3, *SW* required the manual introduction of BGP prefixes in order to group hosts and profile their behaviour. To check if that factor had any impact on how SW was performing, we decided to investigate the

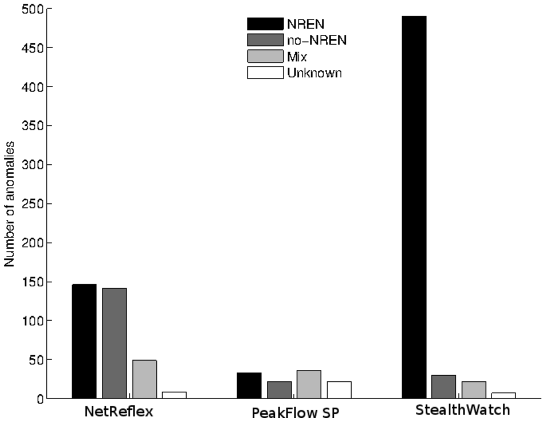


Figure 5: Source of the detected anomalies

origin of the anomalies. Figure 5 shows that, while the sources of the anomalies detected by *PF* and *NR* are spatially split among NRENs and no NRENs, *SW* shows a bias towards NREN origins, the ones for which DANTE was able to provide the BGP prefixes (NRENs are DANTE's customers). The reason behind this is that DANTE could not create those prefixes for its commercial peers due to their enormous variability and size. Therefore, *SW* failed to profile those sources and detect anomalies coming from there. The sources labeled as *Mix* stand for those attacks coming from multiple origins that could not be classified either as NRENs or no-NRENs, because they were a mixture of both of them. However, this group was not significant enough to change the bias showed by *SW*. When we were not able to determine the location of an anomaly in particular (e.g., due to IP spoofing), we classified it as an *Unknown.* This group was not significant either for any of the tools.

3.6. Configurability, Scalability and Usability of the Tools

Besides analyzing the performance of the tools, network operators are also interested in other important features, such as their configurability, scalability and usability. This section shows a qualitative analysis of such features because we think this can be almost as important as the evaluation itself for operators or any other organization with a large network willing to deploy an anomaly detection system in a similar scenario. We describe below what we learnt during the configuration phase of the tools as well as how usable they were during the evaluation process.

3.6.1. Configurability

NR and *PF* were running on one server each, whereas *SW* required a main server and management workstation. All three solutions were deployed in a central location within the GÉANT network (Frankfurt). All tools met requirement 1 (see Section 2.2), i.e., they worked with the already existing NetFlow scenario of GÉANT. *PF* and *NR* required also SNMP access to the routers for obtaining network configuration information. *NR* and *PF* required as well receiving live BGP feeds from our routers. We configured them both to be part of the iBGP (internal BGP) full mesh of our 18 routers. Note that *NR* uses BGP to build a POP to POP traffic matrix (with NetFlow only it is possible to derive the ingress POP, but not the egress one). *PF* only used BGP information for doing traffic peering analysis. As explained in Section 2.3.3 and 3.5, *SW* required the manual feed of static BGP prefixes to cluster the IPs of observed traffic in groups. That was easy for DANTE's customers (European NRENs), whose prefixes are fixed or vary with a very low dynamic, but not feasible for the rest due to their large size and higher dynamism. *NR* also required the collection of IS-IS data, and this was achieved by simply letting the tool server be layer 2 adjacent with one of the routers. This information was not used for the Anomaly Detection component.

3.6.2. Scalability

All the tools proved to be able to handle and process the NetFlow and other data feeds they needed. In particular, *SW* proved to be very accurate for the detection of malicious activity originating/targeting only the set of BGP prefixes defined in advance. However, we estimated that extending it to all our peers (including those providing global connectivity - Telia and Global Crossing) would have required a twenty fold increase in the memory requirements of the tool, thus boosting its cost and impacting its performance.

3.6.3. Usability

Both *SW* and *NR* provided a compact exporting of the information for the detected anomalies. *SW* (being host behavioural based) can also show other anomalies associated to one IP that is the source or target of an anomaly. This functionality is not present in *NR*. However, *NR* can precisely associate an anomaly to an entry/exit point of the network, and to an entry/exit BGP peer due to its fusion of NetFlow and BGP data. *SW* is less precise in this respect, especially when at least one of the anomaly entry/exit points does not belong to a European NREN (which is the most common case). Regarding *PF*, external third-party tools were needed in some cases in order to investigate an event, which made this tool less usable than the others.

3.6.4. Learning Curve

The interface of *NR* is very intuitive as it clearly defines the different available sections (e.g., anomaly detection and traffic analysis). The system has a very short learning curve as the drill depth is approximately six clicks. The interface of *SW* was also very straight forward. The user console is very intuitive, providing multiple paths to investigate an event. At the same time, this could be an issue as the route to the solution could be ten to fifteen clicks before reaching the NetFlow level. *PF* was the least intuitive tool due to the non-practical graph styles and the way to show events back to the operator, which made it harder to analyze each anomaly.

3.7. Tool Selection

Taking into account the requirements listed in Section 2.2 and after evaluating the three tools, *NR* was finally deployed in the GÉANT backbone network. Next, we provide a summary of this evaluation and explain the step-by-step reasoning towards our final selection.

Although it is not possible to determine how it affected their performance, all tools were able to work with sampled input (Sampled NetFlow) (requirement 1). Two of them, *NR* and *PF*, required no additional hardware other than a server running the software and no complex changes to the routers were necessary (requirement 2). However, *SW* needed an extra workstation besides the server and also required the manual introduction of BGP prefixes, which for DANTE's case was only possible for the European NRENs, a subset of all their peers. The performance of the tools in terms of both accurate detection and classification (requirement 3) was surprisingly different and a discriminating factor for the selection process. In terms of true and false positives (Section 3.3), *NR* showed the best results with 82.26% of TP and 15.98% of FP, followed closely by *SW*. *PF* was the worst in that respect. Regarding the types of anomalies detected (Section 3.2), while the *PF* did not report any *Network Scan*, both *SW* and *NR* flagged anomalies of all types. However, when looking at the origin of the attacks (Section 3.5), it was clear that *SW* showed a huge bias towards those anomalies coming from those previously given subset of BGP prefixes (mainly European NRENs), therefore being far less competitive than *NR*, which provided precise identification of the anomalies irrespective of the peering type. As regards the collection of evidence related to an anomaly (requirement 4), *NR* was the one providing the highest detail for a reported anomaly, including entry and exit points in GÉANT and related IPs and ports. Finally, regarding the scalability of the tools (requirement 5), *SW* was the only one that presented issues. In case all BGP prefixes could have been provided, it would have needed an unrealistic amount of memory.

All in all, due to its easy configuration, its best detection and classification, its independence of the origin of the anomaly, its scalability and its higher detail for a reported attack, *NR* was clearly the best tool given DANTE's requirements, and, therefore, the software finally deployed in GÉANT.

3.8. Discussion

It must be noted that the tool finally deployed in GÉANT, *NR*, is merely anecdotal. The relevance of this part of our work for other network operators or any other sort of organization lies on the followed methodology and the performed experiments during the benchmarking rather than on the final decision, which will always depend on the particular network environment and specific needs.

For instance, for *SW*, it is clear that the impossibility to provide all the BGP prefixes (for both size and scalability issues), significantly reduced the detection capabilities of the tool. In a different scenario, where the organization willing to deploy it could provide this information, it could be perfectly possible that *SW* outperformed *NR*.

Regarding *PF*, although it showed the worst overall performance, it was the best tool detecting *Denial-of-Service* attacks, with almost no false positives (both *SW* and *NR* reported significant proportions of FP for this particular anomaly). Moreover, *PF* is the only solution providing mitigation of an attack after detection. To give an example, for an ASP (application service provider), which is interested in assuring high quality and availability of its services, this solution would fit better than the others because it flags *DoS* with high accuracy and, additionally, is capable of blocking the malicious traffic while allowing legitimate users to continue using the service.

Please note that, even though *PF* performed poorly in GÉANT, according to its manufacturer, Arbor Networks, *PF* is one of the most widely deployed commercial solutions for anomaly detection. Therefore, this confirms the fact that a particular software is neither good nor bad by itself, but depends on its adequacy to the network environment and the singular requirements of the operator.

4. Analysis of the Anomalies

After selecting the set of tools to analyze (Section 2) and evaluating them (Section 3), *NetReflex (NR)* was deployed in the GÉANT backbone network. In this section, we present a study about the anomalies we have found after operationally using this tool for half a year. We show their their types, properties, magnitudes, origins and destinations.

4.1. Validation of the Deployed Tool

This study is based on another dataset labeled as *dataset-2* (see details in Table 1). While *dataset-1* was obtained during the evaluation of the three tools, *dataset-2* was collected using the deployed tool (*NR*). *dataset-2* covers almost a seven months period (10^{th} of November 2009 - 3^{rd} of May 2010). In order to confirm that the tool was performing as expected, since it was not feasible to manually check all the anomalies in *dataset-2* due to its duration, a subset of six days of NetFlow data (10^{th}, 11^{th}, 16^{th}, 17^{th}, 23^{rd} and 26^{th} of November 2009) was collected (see details in Table 3). All these anomalies were manually validated following exactly the same methodology explained in Section 3.1.

In Table 3 we can observe that the TP-FP (88.01%-11.99%) of *NR* improved with respect to its TP-FP during the tool evaluation period (82.26%-15.98%, Section 3). The main reason behind this difference are two key modifications made to *NR* after its deployment in GÉANT. In particular, the changes made by the vendor were the following. Firstly, the traffic in case of UDP floods was systematically checked in the reverse direction of the attack (i.e., outgoing from the target). This check was motivated by the fact that, during the evaluation period, it was observed that most of the false positives for *DoS* were actually

Table 3: Details for the 6 manually analyzed days of *dataset-2* (November 2009).

Day	#TP	#FP	#flows	#packets	#bytes
10th	39	9	619.43M	4.09G	2.15T
11th	40	4	613.06M	4.01G	2.10T
16th	44	4	581.34M	3.55G	1.99T
17th	38	2	604.66M	3.73G	2.06T
23rd	45	9	598M	3.45G	2.03T
26th	29	4	560.06M	3.67G	2.01T
Overall	235	32	3.57G	22.5G	12.33T

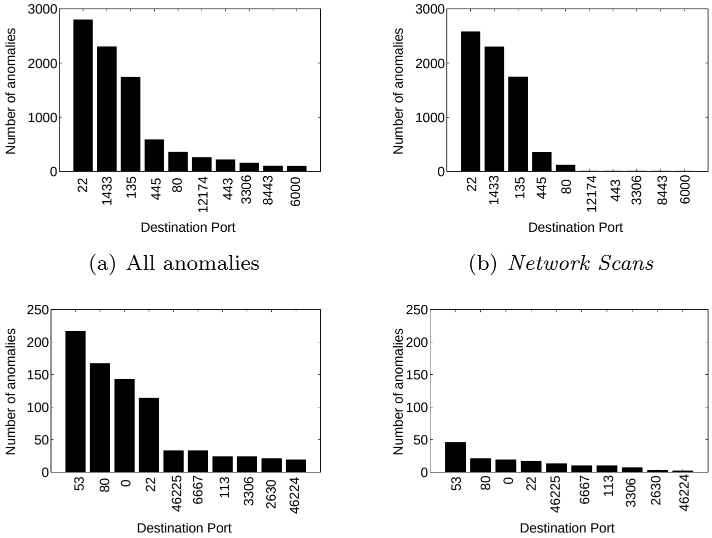
bandwidth tests, large transfers, high-volume P2P activity and data streaming, all of which require minimal bidirectional interaction between the involved hosts. Consequently, if the reverse portion of the traffic turned out to be more than a given threshold of the incoming, it was assumed that there was a legitimate communication going on and no anomaly was signaled. We selected a threshold of 10%, which resulted in a good tradeoff between TP and FP for *DoS*. This change significantly reduced the overall *FP*. Secondly, the sensitivity of the algorithm towards commonly attacked ports was improved, as it will be explained in Section 4.2. Also, recall that *dataset-1* was captured under 1/1000 sampling while *dataset-2* was collected with a sampling rate of 1/100, which might contribute as well to the overall performance improvement of *NR*.

4.2. Anomaly Distribution

As already observed in *dataset-1*, *dataset-2* confirmed that *Network Scans* are clearly the most frequent attack with a percentage of 79%, while the rest of the considered security attacks (*Port Scans*, *DoS* and *DDoS*) represent the remaining 21%. It seems hat *NS* are some sort of background activity that is almost always going on looking for open well-known ports to later on exploit some known vulnerability associated to services running on these ports. *DoS* and *Port Scans* are the next most frequent attacks with a quite similar percentage (11% and 8% respectively). The least common are *DDoS* attacks with only 2% of the reported anomalies.

The reason why the percentage of detected *NS* changed so much between *dataset-2* (79%) and *dataset-1* (42%, see Figure 2) is, as already mentioned in Section 4.1, because *NR* was tuned after its deployment in GÉANT. The vendor of *NR* added the capability to select, at configuration time, specific destination ports where to increase the anomaly analysis sensitivity. We used that feature to make it focus on frequently attacked ports that were reported by *StealthWatch (SW)* but missed by *NR* during the evaluation. Recall that, according to Figure 4, *SW* detected more than twice as many *NS* as *NR* (before tuning). In particular, we added ports 22 (SSH), 135 (RPC), 139 (Netbios), 445 (SMB) and 1433 (SQL), which are often misused (e.g., SSH brute force attempts or SQL injections) and thus enhanced the detection of the algorithm.

We think that the anomalies detected by *NR* after the tuning better reflected the reality than before modifying it. The anomaly distribution of *NR* after



(a) All anomalies

(b) *Network Scans*

(c) *DoS*

(d) *DDoS*

Figure 6: Top attacked ports

tuning (*NS*:79%, *PS*:8%, *DoS*:11%, *DDoS*:2%) is more similar to *SW* (*NS*:72%, *PS*:2%, *DoS*:17%, *DDoS*:<1%). However, note that the distribution of the anomalies found in the GÉANT backbone network could still be biased towards how good or bad is *NR* in detecting each type of attack.

4.3. Top Attacked Ports

Looking at Figure 6(a) we can see the top 10 ports attacked taking into account all the anomalies. We observed that seven of them were attacks to well-known ports of widely-known services: port 22 (SSH), port 1433 (Microsoft SQL), ports 135 and 445 (Windows) and port 80 (Web) are in the top 4, while port 443 (HTTPS) and 3306 (MySQL) are in the 7th and 8th position. The other three most attacked ports were 12174, 8443 and 6000. The first port refers to a vulnerability that affects outdated Symantec servers from fall 2009. The second is a popular non-standard alternative for listening to HTTPS connections, and the third is used in X-Window servers. Given that this only shows the top attacked ports regardless of the attack, we then analyzed how that distribution looked like for every type of anomaly in order to see if there are particular ports preferred for each kind of attack.

We observed that, as expected, the overall top targeted ports was indeed dominated by the main ports attacked for *Network Scans* (see Figure 6(b)), which is expected given that it is, by far, the most frequent attack. The most common subtype is SSH scanning, followed by port 1433 and Windows Netbios ports for networking functions such as file-sharing (ports 135 and 445). Port 80 is the next one, but with lower intensity. Regarding the five ports left (12174, 443, 3306, 8443 and 6000), they all had less than 10 instances during these analyzed period.

Regarding *DoS* (Figure 6(c)), the most significant attacks happened on ports 53, 80, 0 and 22. Attacks to DNS ports are normally DNS cache poisoning attempts, which means that a DNS server has received an update from a non-authoritative DNS source and thus its clients are receiving fake data. Regarding attacks to ports 80 and 22, they are quite well-known and intend to saturate either a web or a SSH server to make it unavailable to its clients. We then found port 0, which according to IANA [15] is reserved, but sometimes is used in OS fingerprinting activities. However, we believe that the majority of "port zero" flows are due to packet fragmentation, and to the fact that NetFlow v5 creates a flow with destination port zero for fragmented packets. As showed in Figure 6(d), for *DDoS*, port 80 is clearly the most targeted port. The second one is port 6667 (IRC or Internet Relay Chat), which is often used to remotely control hosts previously infected by a Trojan (zombies). These set of hosts are called "Botnets" and can be used to launch massive *DDoS* attacks. We finally found the well-known attacks to ports 53 and 22. The set of remaining ports were 25345, 2001 (Trojan), port 0 again and 7000 (Trojan).

4.4. Magnitude of the Anomalies

In this section, we provide an analysis about the average number of flows, packets and bytes involved in each sort of anomaly. Given that *NR* does not provide volume information together with the reported anomalies, we used the raw NetFlow data saved for the subset of six days that we manually validated. In order to obtain the correct flows associated to each anomaly, we used the same method described in Section 3.4, a recent extension [14] of the Apriori algorithm [13] .

Table 4 shows that the volumes associated to each sort of anomaly are significantly different. Concerning the number of flows, it is clear that *DoS* attacks rarely use more than one or two flows, while the other anomalies involve a much higher number. The anomaly type using more flows with a huge difference is the *Distributed DoS*. We then find the scans: in the first place there is the *Port Scan*, with a large number of flows corresponding to the different ports tested and then we clearly observe that the intensity of a *Network Scan* is the lowest (in terms of flows). Regarding the number of packets, the average for *DDoS* and *DoS* is the highest while *NS* are, by far, the attacks using the least. Both *Network Scans* and *DDoS* generate single-packet flows (they show equal amount of packets and flows, respectively). Finally, regarding the number of bytes, *Port Scans* do not seem to be "stealthy" activities and have a number of bytes comparable to *DoS* even that being lower. *NS* are the ones using the lowest amount of bytes. Regarding *DoS* and *DDoS* we can see that they look almost exact in terms of packets and bytes but they clearly differ concerning the amount of flows they use for attacking: *DoS* use a few flows while *DDoS* launch attacks coming from a huge amount of sources.

Table 4: Average number of flows/packets/bytes per anomaly type found in the six manually validated days inside dataset-2

	flows	packets	bytes
Network Scans	1.75K	1.75K	74.8K
Port Scans	153.47K	347.33K	9.64M
DoS	2.33	960.22K	40.10M
DDoS	1.15M	1.15M	46.06M

4.5. Origin and Destination of the Anomalies

This section presents a study of how the origins and the destinations of the anomalies were distributed over GÉANT. Most of them (56%) were generated from outside GÉANT and 38% of them came from the inside. The remaining 6% came from an unknown location (see end of Section 3.5 for more details). Regarding the destination, a huge amount of the attacks (70%) were directed to the GÉANT network while 24% of the anomalies had outside targets and 6% had unidentified receivers.

Surprisingly, almost half of all the anomalies analyzed (45%) came solely from the Asia-Pacific region, specially from China, probably because of the high amount of infected PCs running non-properly patched Windows OS. Regarding the remaining countries, none of them generated (separately) more than 6% of the overall anomalies. Concerning the top targets, there was not a clearly predominant region. Israel (not a EU member but connected to GÉANT) was the most commonly attacked country (8% of the anomalies) along with Greece (7%), North America (6%), Portugal (5%) and Estonia (5%). Israel and Estonia have had and still have some political issues that may explain their presence at the very top even though they are small networks (they receive/send little traffic with respect to other bigger research networks).

When studying the top origin-destination pairs, we found that the most frequent one was from no-NREN to NREN (53.38%). This was due to the fact that the most frequent type of anomaly, as we saw in Section 4.2, was the *Network Scan*, which we can consider as some sort of background activity proportional to the number of hosts in a network. Since the no-NRENs represent the "rest of the Internet" (i.e. non-academic networks) it was quite expected this number of anomalies to be numerically dominant. The next pairs were from NREN to no-NREN, with a far lower percentage (21%), and from NREN to NREN (16.78% of the anomalies). The remaining pairs represented less than 9% of the attacks.

5. Background

Although the algorithms used by the tools evaluated are proprietary and therefore we cannot know exactly how they work, in this Section, we briefly explain the different anomaly detection approaches that they are based on. We refer the interested reader to [16] for further information about each particular anomaly detection technique described below.

PCA-based. *NetReflex* is based on this approach. The subspace-method [11, 12] analyzes OD-flows (flows with the same origin and destination points of the monitored network). Because of the high dimensional multivariate data structure of that flows, a lower-approximation is needed: Principal Component Analysis (PCA) [17]. This mathematical method captures the most important trends of the explored data (it preserves the significance of the data while reducing its complex initial structure). Then, the subspace method splits the resulting output into normal and anomalous. If the projection of the data in the second space is higher than a previously given value, an anomaly is flagged.

Statistical-based. *PeakFlow SP* is mainly based on this technique. Mechanisms in this group use time series prediction[18–20] to make estimations about the future value of the monitored variable taking into account its historical evolution. If the real value of the measured metric differs too much from its prediction, it is considered to be an anomaly. There are several approaches to do the prediction. *Exponential Smoothing* [18] is the simplest method: the next value is the average between the last prediction and the current real value. The problem of this approach is that it does not account for seasonality, so it would always report anomalies due to changes because of normal activity patterns (e.g., day vs. night traffic). Another well-known prediction model is called *The Holt-Winters Forecasting Algorithm* [18] and it tries to overcome the previous problem. Its prediction is an average of three variables that account for baseline, linear trend and seasonality respectively. Each of these variables is updated using the *Exponential smoothing* technique explained above.

Signature-based. *PeakFlow SP*, besides using statistical-based anomaly detection, also uses this approach. Signature-based methods rely on a database of patterns related to already discovered attacks. When there is a match with one of those patterns (known as signatures), the system triggers an alarm. There are well-known and widely used solutions such as Snort [21] and Bro [22] Intrusion Detection Systems based on this. However, while Snort is solely based on signatures, Bro also does a more complex analysis of the data. The main downside of all tools using only signature-based anomaly detection is that they miss the new attacks since they are only able to detect incoming threats matching pre-stored patterns.

Behavior-based. *StealthWatch* belongs to this group. Techniques inside this approach perform behavior analysis (see [23, 24] and references therein). The key idea is to profile the normal behavior of a host and detect when it changes. After building the per-user profile, it can be easily detected if its behavior differs from its usual activities. The main drawback of this approach is the necessity to keep the state for each speaking host, which might be especially problematic for large networks.

6. Conclusions

In this paper, we analyzed three commercial tools for anomaly detection and provided a study about the type and characteristics of the current security threats happening in a large backbone network. We also reported the strengths and shortcomings found while using these tools, as well as the experience and knowledge we acquired during this long process.

After manually classifying more than 1000 anomalies, we learnt that their distribution, as well as the accuracy of each tool, were significantly different. While the true positives were generally reasonable, the ratio of false positives was quite high for all tools. Surprisingly, we found that the overlap between the anomalies detected by different tools was minimal. This indicates that the number of false negatives is still significant even in commercial tools, and shows the importance of combining different approaches to obtain a stronger anomaly detection system by potentially catching a broader range of malicious events. As for what tool performed better for each anomaly type, we observed that while *StealthWatch* (based on host behaviour) was the best with *Network Scans*, *PeakFlow SP* (based on traffic volumes) was better at discovering *DoS*. *NetReflex* (based on PCA and entropy) exhibited the best balance regardless of the anomaly type.

In addition, we studied the most common types of anomalies, the top attacked ports, the volumes associated to each anomaly and their sources and destinations after using the deployed tool for approximately six months. Our study revealed the tremendous frequency and persistence of *Network Scan* attacks. We also showed that every type of anomaly had its own preferred destination ports. As expected, the overall top-3 is governed by well-known targets: SSH (22), Microsoft SQL (1433) and a Windows resource-sharing (135). Regarding the magnitude of each sort of anomaly, while *DoS* rarely use more than 1 or 2 flows, *DDoS* attacks generate, by far, the highest amount of flows. *Network Scans* involve the lowest number of packets and bytes and few flows. On the contrary, *Port Scans* use a quite large number of flows, packets and bytes. Moreover, we observed that the Asia-Pacific region turned out to be the region generating most of the attacks while small countries like Israel or Estonia were common targets.

From a practical point of view, we also reported on the acquired experience during this long process. We realized about the complexity and diversity of the traffic observed in a large academic network. For instance, it is quite common to observe traffic that behaves like a *Denial-of-Service* but happens to be some legitimate research experiment. This fact points out a key learning aspect that should be central for any anomaly detection tool. Each network is different and the detection algorithm must be flexible enough to adapt to it. Moreover, the configuration of the tools as well as their long term scalability are important aspects. To give an example, while *StealthWatch* would have required substantial upkeep in maintaining the prefix lists, *NetReflex* did not need anything from a manual perspective.

Acknowledgements

This work was supported by two STSM grants (refs. 5202 and 5663) from the TMA-COST Action *IC0703 "Data Traffic Monitoring and Analysis"* and also by the Spanish Ministry of Science and Innovation under contract TEC2011-27474 (NOMADS project) and by the *Comissionat per a Universitats i Recerca del DIUE de la Generalitat de Catalunya* (ref. 2009SGR-1140).

References

[1] Cisco Systems, NetFlow services and applications, White Paper (2000).

[2] GÉANT, http://www.geant.net.

[3] DANTE, http://www.dante.net.

[4] Cisco Systems, Sampled NetFlow, http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html.

[5] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, A. Lakhina, Impact of packet sampling on anomaly detection metrics, in: Proceedings of IMC, 2006.

[6] J. Mai, C. Chuah, A. Sridharan, T. Ye, H. Zang, Is sampled data sufficient for anomaly detection?, in: Proceedings of SIGCOMM, 2006.

[7] J. Mai, A. Sridharan, C. Chuah, H. Zang, T. Ye, Impact of packet sampling on portscan detection, IEEE Journal on Selected Areas in Communications 24 (12) (2006) 2285–2298.

[8] Guavus, *NetReflex*, http://www.guavus.com.

[9] Arbor Networks, *Peakflow SP*, http://www.arbornetworks.com/peakflowsp.

[10] Lancope, *StealthWatch*, http://www.lancope.com/products/StealthWatch-System.

[11] A. Lakhina, M. Crovella, C. Diot, Characterization of network-wide anomalies in traffic flows, in: Proceedings of IMC, 2004.

[12] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: Proceedings of SIGCOMM, 2005.

[13] D. Brauckhoff, X. Dimitropoulos, A. Wagner, K. Salamatian, Anomaly extraction in backbone networks using association rules, in: Proceedings of IMC, 2009.

[14] I. Paredes-Oliva, X. Dimitropoulos, M. Molina, P. Barlet-Ros, D. Brauckhoff, Automating root-cause analysis of network anomalies using frequent itemset mining, ACM SIGCOMM Computer Communication Review 40 (4) (2010) 467–468.

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

[16] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys 41 (3) (2009) 15.

[17] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, N. Taft, Structural analysis of network traffic flows, in: Proceedings of SIGMETRICS, 2004.

[18] P. Brockwell, R. Davis, Introduction to time series and forecasting, 1996.

[19] J. Brutag, Aberrant behavior detection and control in time series for network monitoring, in: Proceedings of LISA, 2000.

[20] H. Wang, D. Zhang, K. Shin, Change-point Monitoring for the detection of DoS attacks, IEEE Transactions on Dependable and Secure Computing (2004) 193–208.

[21] M. Roesch, Snort–lightweight intrusion detection for networks, in: Proceedings of USENIX Systems Administration Conference, 1999.

[22] V. Paxson, Bro: a system for detecting network intruders in real-time, Computer Networks 31 (23-24).

[23] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection, in: Proceedings of the Third SIAM International Conference on Data Mining, 2003, pp. 25–36.

[24] MINDS, Minnesota Intrusion Detection System, http://www.cs.umn.edu/research/MINDS/.