

What is the Impact of P2P Traffic on Anomaly Detection?^{*} ^{**}

Irfan Ul Haq, Sardar Ali, Hassan Khan and Syed Ali Khayam

School of Electrical Engineering & Computer Science
National University of Sciences & Technology (NUST)
Islamabad 44000, Pakistan

{irfan.haq,sardar.ali,hassan.khan,ali.khayam}@seecs.nust.edu.pk

Abstract. Recent studies estimate that peer-to-peer (p2p) traffic comprises 40-70% of today's Internet traffic [1]. Surprisingly, the impact of p2p traffic on anomaly detection has not been investigated. In this paper, we collect and use a labeled dataset containing diverse network anomalies (portscans, TCP floods, UDP floods, at varying rates) and p2p traffic (encrypted and unencrypted with BitTorrent, Vuze, Flashget, μ Torrent, Deluge, BitComet, Halite, eDonkey and Kademlia clients) to empirically quantify the impact of p2p traffic on anomaly detection. Four prominent anomaly detectors (TRW-CB [7], Rate Limiting [8], Maximum Entropy [10] and NETAD [11]) are evaluated on this dataset.

Our results reveal that: 1) p2p traffic results in up to 30% decrease in detection rate and up to 45% increase in false positive rate; 2) due to a partial overlap of traffic behaviors, p2p traffic inadvertently provides an effective evasion cover for high- and low-rate attacks; and 3) training an anomaly detector on p2p traffic, instead of improving accuracy, introduces a significant accuracy degradation for the anomaly detector. Based on these results, we argue that only p2p traffic filtering can provide a pragmatic, yet short-term, solution to this problem. We incorporate two prominent p2p traffic classifiers (OpenDPI [23] and Karagiannis' Payload Classifier(KPC) [24]) as pre-processors into the anomaly detectors and show that the existing non-proprietary p2p traffic classifiers do not have sufficient accuracies to mitigate the negative impacts of p2p traffic on anomaly detection.

Given the premise that p2p traffic is here to stay, our work demonstrates the need to rethink the classical anomaly detection design philosophy with a focus on performing anomaly detection in the presence of p2p traffic. We make our dataset publicly available for evaluation of future anomaly detectors that are designed to operate with p2p traffic.

1 Introduction

During March of 2009, a record number of 4,543 anomalies was recorded by an open-source TRW-CB based [7] anomaly detector deployed on our school's

^{*} The original publication is available at <http://www.springerlink.com>

^{**} This work is supported by Pakistan National ICT R&D Fund.

network. The network administrators took it as a result of a zero-day attack and updated the antivirus and antispysware definitions on school hosts. However, TRW-CB continued reporting anomalies even after the update. An investigation of this strange behavior by correlating the TRW-CB logs and the network logs revealed that the culprit was p2p traffic which was being reported as anomalous.¹ This strange behavior of TRW-CB was communicated to us which intrigued us to investigate the impact of p2p traffic on anomaly detection.

Based on the results of our investigation, in this paper we empirically answer the following open question: *How much perturbations are introduced in anomaly detection metrics by p2p traffic² and how can these perturbations be mitigated?* A general answer to this question can be inferred intuitively because some features of p2p traffic are quite similar to those of malicious traffic and quite different from the bulk of benign TCP traffic [5]. Hence, the accuracy of an anomaly detector, which flags deviations from a model of normal behavior, is bound to degrade in the presence of p2p traffic. For example, the decentralized and distributed nature of the p2p architecture results in establishment of a large number of connections to random ports during boot-strap which shares similarities with portscan attacks; compare a torrent client “scanning” over 50 peers during bootstrapping to MyDoom-A with an average scan rate of 9 scans per minute. Similarly, high churn rate in p2p networks results in a large number of failed connections³ which is another commonly-observed phenomenon during portscan attacks.

While a general sense can be determined intuitively, our empirical study gives deeper insights by breaking the above question into the following set of important sub-questions: 1) How much degradation does p2p traffic induce in anomaly detection accuracy (detection and false positive rates)? 2) Which anomaly detection metrics/principles are more sensitive to p2p traffic and why? 3) Does the aggressive nature of p2p traffic dominate some/all attack classes and high-/low-rate attacks? 4) Can an anomaly detector handle p2p traffic if it is trained on a dataset containing p2p traffic? 5) Can a pragmatic solution be designed to make an anomaly detector insensitive to the p2p traffic? 6) Can existing public p2p traffic classifiers mitigate the degradation in anomaly detection accuracy? 7) What are the open problems in designing anomaly detectors which operate effectively in today’s Internet traffic?

To empirically answer the above questions, we collect a labeled dataset containing diverse network anomalies (portscans, TCP floods, UDP floods, at varying rates) and p2p traffic (encrypted and unencrypted with BitTorrent, Vuze, Flashget, μ Torrent, Deluge, BitComet, Halite, eDonkey and KAD clients). Since it is not possible to evaluate all existing anomaly detectors, ROC-based accu-

¹ This sudden spike was caused by recent relocation of students’ dormitories inside our newly-built campus and the students’ usage of p2p applications in their dormitories.

² While our evaluations focus on p2p file sharing traffic, p2p VOIP and p2p streaming video traffic also exhibit similar traffic behaviors.

³ Failed connections is a feature which is employed to detect malicious hosts [6]-[9] as well as p2p file sharing hosts [20],[21].

racies of four prominent anomaly detectors (TRW-CB [7], Rate Limiting [8], Maximum Entropy [10] and NETAD [11]) are evaluated on this dataset.

Our results reveal that *all* the anomaly detectors experience an unacceptable (up to 30%) drop in detection rates and a significant (up to 45%) increase in false alarm rates when operating with p2p traffic. Henceforth in the paper, we refer to this accuracy degradation as the *torrent effect* on anomaly detection. We evaluate the torrent effect by evaluating the anomaly detectors on different attack rates and classes. We show that anomaly detectors deliver varying accuracies on different attack classes and this varying performance is a function of the design principle of a given anomaly detectors. Similarly, we show that p2p traffic inadvertently acts as a very effective evasion cover for low-rate attacks as detection of such attacks is seriously affected by p2p traffic.

Based on the significant and consistent accuracy degradations observed in our study, we argue that a p2p traffic classifier based pre-processor can offer the anomaly detectors a pragmatic, albeit short-term, relief from the torrent effect.⁴ By incorporating OpenDPI [23] into the IDSs we see 12% improvement in detection accuracy with 4% reduction in false positive rate. Similarly, incorporating KPC [24] results in 18% improvement in detection accuracy and a 48% reduction in false positive rate. However, even with these improvements, existing non-proprietary p2p traffic classifiers do not have sufficient traffic classification accuracies to eliminate the torrent effect.

Recent trends indicate that the volume of p2p traffic is reducing as service providers are now deploying commercial p2p traffic classifiers to throttle p2p traffic in real-time [1]-[4]. Nevertheless, due to the ubiquity and popularity of p2p networks and software, even with reduced-volumes, p2p traffic is anticipated to continue comprising a significant percentage of the Internet's traffic in the coming years [34]. We therefore advocate a fundamental rethinking of the anomaly detection design philosophy with future anomaly detectors catering for p2p traffic in their inherent design. We make our dataset publicly available for evaluation of such future anomaly detectors.

2 Related Work and Background

While significant research has recently been focused towards evaluating and understanding trends in anomaly detection [16], to the best of our knowledge, the impact of p2p traffic on intrusion detection has not been explored. Therefore, in this section we only provide a brief overview of the anomaly detectors evaluated in this work; interested readers are referred to the original papers [7],[8],[10] and [11] for detailed descriptions of the anomaly detectors.

The Rate Limiting approach [8], detects anomalous connection behavior by putting new connections exceeding a certain threshold in a queue. An alarm is raised when the queue length exceeds a threshold. TRW-CB [7] limits the rate at which new connections are initiated by applying the sequential hypothesis

⁴ Commercial IDSs are already incorporating p2p traffic classifiers (DPI engines) into their products [31]-[33].

testing and by using a credit increase/decrease algorithm to slow down hosts that are experiencing unsuccessful connections. The Maximum Entropy based detector [10] estimates the benign traffic’s baseline distribution using Maximum Entropy method by dividing the traffic into 2,348 packet classes. These packet classes are defined based on destination ports and the transport protocols. Kullback-Leibler (K-L) divergence measure is then used to flag anomalies if divergence from the baseline distribution exceeds a threshold from the baseline distribution. NETAD [11] operates on rule-based filtered traffic in a modeled subset of common protocols. It computes a packet score depending on the time and frequency of each byte of packet, and rare/novel header values are assigned high scores. A threshold is applied on a packet’s score to find anomalous packets. For performance evaluations, parameter tuning for these anomaly detectors is performed in the same fashion as in a recent evaluation study [35].

We chose these anomaly detectors to ensure diversity because these detectors have very different detection principles and features, and operate at different traffic granularities. On the one hand, we use Rate Limiting [8] which is a connection-based programmed system using a thresholding approach, while, on the other hand, we use a statistical programmed system, TRW-CB [7]. Similarly, we employ an information-theoretic self-learning system like Maximum Entropy [10] as opposed to NETAD [11] which is a packet-based rule-modeling system.

3 Dataset Description

For the present problem, we wanted to use real, labeled and public background and attack datasets to measure the accuracy. Furthermore, for comprehensive evaluation, we needed attacks of different types (DoS, portscan, etc.) and different rates for each attack type. Finally, we needed labeled p2p traffic from various clients and p2p protocols in our dataset. While some old attack datasets are available [17]-[19], they do not contain p2p traffic and do not contain attacks of different types. Therefore, we collect our own network traffic dataset and make it publicly available for repeatable performance evaluations.⁵ The rest of this section describes our data collection methodology.

We collect dataset in our campus network. The research labs in our campus are located in research wing and traffic from each research lab is relayed through a 3COM4500G switch to research wing’s Cisco 3750 router using fiber connections, as shown in Figure 1. The wing router is connected to the distribution router which handles traffic of the entire campus. The research wing router routes traffic for over 50 computers. Three computers in our research lab were used to generate attack traffic. P2P traffic was generated by hosting p2p file sharing applications on twelve computers in different labs. Due to privacy constraints, we were only allowed to collect traces at the research wing router. We now provide the details for normal, p2p and attack traffic in our dataset.

⁵ The dataset collected for this work is available at <http://wisnet.seecs.nust.edu.pk/projects/ENS/DataSets.html>.

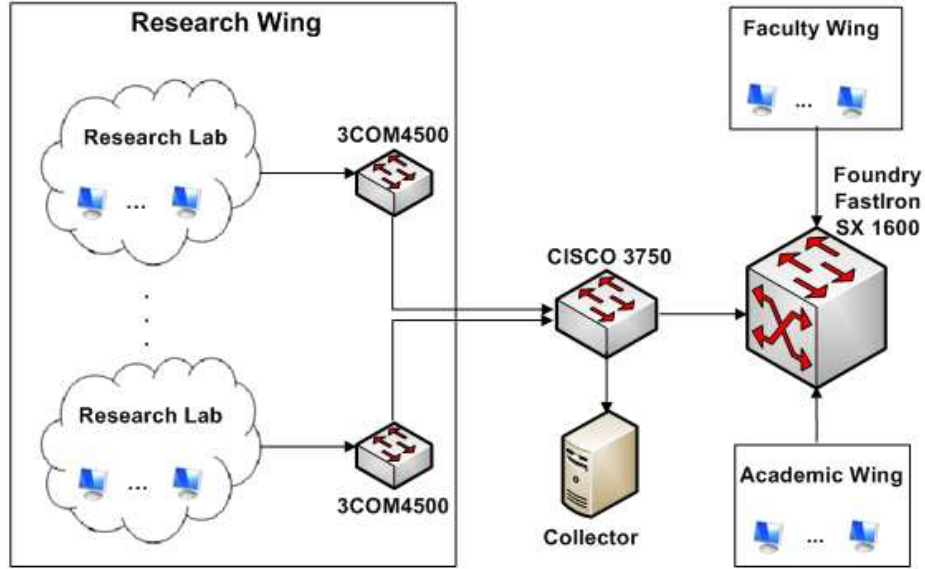


Fig. 1. Dataset collection setup.

Table 1. P2P File Sharing Application Traffic Statistics

| Client Name & Version | Sessions Estb. | Traffic Vol. | Throughput(Mbps) |
|-----------------------|----------------|--------------|------------------|
| Vuze 4.0 | 20 | 685 MB | 0.8 |
| Flashget 1.9.6 | 62 | 60.7 MB | 1.2 |
| UTorrent 1.8.1 | 30 | 1.08 GB | 1.7 |
| BitTorrent 6.1.2 | 134 | 1.59 GB | 2.62 |
| Deluge 1.0.7 | 30 | 171 MB | 0.72 |
| BitComet 1.07 | 20 | 57.4 MB | 0.6 |
| Halite 0.3.1 | 9 | 413 MB | 0.94 |
| eMule v0.49b | 203 | 2.67 GB | 1.2 |

3.1 Normal Traffic

We captured the normal traffic in six periods, each one of over three hours. During traffic capturing, different applications were hosted on the machines including file transfer, web browsing, instant messaging, real-time video streaming, etc. It was ensured that during normal traffic capturing, no p2p application was hosted on any of the client machines. The mean packet rate recorded for the background traffic was about 3168 pkts/sec and the standard deviation was 1683 pkts/sec.

3.2 P2P Traffic

The p2p traffic in our traces belongs to the BitTorrent, eDonkey and Kademlia protocols. These protocols were chosen as representative traffic from p2p traffic

Table 2. Attack Characteristics & Background Traffic Information During Attacks

| Attack Name | Attack Characteristics | Attack Rate (pkts/sec) | Background Traffic Statistics at attack time (pkts/sec) | |
|---------------------|---|------------------------|---|----------|
| | | | μ | σ |
| TCP-SYN portscans | Fixed src IP addr Two distinct attacks: First scan on port 80, Second scan on port 135 | 0.1 | 2462.9 | 474.4 |
| | | 1 | 3002.6 | 398.0 |
| | | 10 | 3325.2 | 397.7 |
| | | 100 | 6100.0 | 2492.4 |
| | | 1000 | 3084.7 | 247.4 |
| TCP-SYN flood (DoS) | Two remote servers attacked Attacked ports: 143, 22, 138, 137, 21 | 0.1 | 2240.1 | 216.7 |
| | | 1 | 2699.1 | 328.8 |
| | | 10 | 4409.8 | 1666.2 |
| | | 100 | 3964.1 | 1670.4 |
| | | 1000 | 3000.9 | 238.0 |
| UDP flood fraggle | Two remote servers attacked Attacked ports: 22, 80, 135, 143 | 0.1 | 2025.8 | 506.4 |
| | | 1 | 2479.1 | 291.0 |
| | | 10 | 4028.4 | 1893.1 |
| | | 100 | 6565.7 | 3006.9 |
| | | 1000 | 2883.7 | 260.8 |

class because these protocols generate the largest volumes of p2p traffic on Internet [1]. During our trace collection for BitTorrent protocol, we used multiple torrent files for transferring data from/to multiple geographical locations for each torrent session. Multiple clients including Vuze, Flashget, μ Torrent, BitTorrent, Deluge, BitComet and Halite were used to introduce real-world diversity in the dataset as different clients might had different behavior. For eMule sessions two options related to protocol obfuscation and communication with obfuscated connections only (“Allow obfuscated connections only”), were enabled in the client to ensure logging of encrypted traffic. Similarly, encryption was enabled for the torrent sessions. Statistics for the p2p file sharing applications’ traffic are given in Table 1.

3.3 Attack Traffic

For attack traffic, we launch port scans (TCP SYN), DoS (TCP SYN) and fraggle (UDP flood) simultaneously from three end hosts in our research lab. The DoS attacks was launched on two servers under our administration with public IP addresses. Each attack was launched for a period of five minutes with spoofed IP address. For every attack type, three low-rate ($\{0.1, 1, 10\}$ pkts/sec) and two high-rate ($\{100, 1000\}$ pkts/sec) instances were launched. The attack characteristics for each attack are shown in Table 2.

4 Investigating the Torrent Effect

We now embark on finding answers to the questions that were raised in the introduction. To this end, we evaluate the anomaly detectors on datasets with varying proportions of attack and p2p traffic. In this section, we perform the evaluations to find out the impact of p2p traffic on anomaly detection accuracy; its correlation with high- and low-rate attacks; its affect on different attack classes and whether p2p traffic should be used to train an anomaly detector. We defer the solution to the torrent effect to Section 5.

4.1 How much degradation does p2p traffic induce in anomaly detection accuracy?

We first investigate the impact of p2p traffic on the anomaly detectors' detection and false alarm rates. Figure 2 plots the Receiver Operating Characteristic (ROC) curves of the anomaly detectors on the dataset with p2p traffic and on the same dataset with p2p traffic removed from it. The anomaly detectors in this case were trained only on non-p2p traffic. With the introduction of p2p traffic, the detection rates of all anomaly detectors drop and the false positive rates increase. This behavior is observed because of the similarities between p2p and malicious traffic features, such as a large number of connection attempts, a large number of failed connections, and the use of unprivileged ports. Figure 2 shows that Maximum Entropy and TRW-CB fail miserably (up to 30% reduction in detection rate and up to 40% increase in false positives) when they operate on the dataset with p2p traffic. On the other hand, the detection rates of Rate Limiting and NETAD never degrade by more than 20% and 10%, respectively. Similarly, for Rate Limiting and NETAD, the average false positive rate increase remains around 10%. Deferring further discussion on relative degradation for each anomaly detector to the next section, we deduce from Figure 2 that the accuracies of *all* anomaly detectors degrade considerably due to p2p traffic.

4.2 Which anomaly detection metrics/principles are more sensitive to p2p traffic and why?

As we discussed in Section 2, we chose a diverse set of anomaly detectors which employ varying traffic features and operate on assorted detection principles. We now analyze the sensitivity of each detector to p2p traffic with a motivation to identify design guidelines to make these detectors insensitive to background traffic.

Figure 2 shows that NETAD provides the best overall accuracy and sustains it under p2p traffic. This is surprising because NETAD is in essence a rule-based detector and previous studies have shown that such algorithms fail in many attack scenarios [35, 36]. Further investigation revealed that the graceful accuracy degradation of NETAD is mainly because of two rules that it uses to classify normal traffic: 1) All UDP traffic on higher ports (> 1023); 2) TCP data starting after 100 bytes. Both of these rules are satisfied by most of the

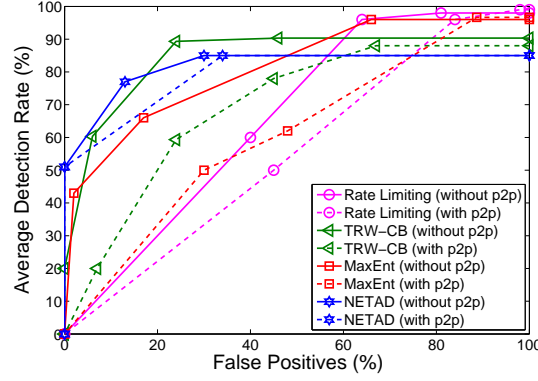


Fig. 2. ROC results to quantify the impact of p2p traffic on anomaly detection accuracy; each ROC point is averaged over $3(\text{attacks}) \times 3(\text{instances/attack}) \times 5(\text{rates/instance}) = 45$ attack windows of 5 minutes each.

p2p clients because the communication with trackers and peers takes place on higher ports, and TCP connections with each peer requires a sequence of TCP control packet exchanges to establish the number and sizes of file chunks to be downloaded. Due to these rules, NETAD continued to detect most of the p2p traffic as non-malicious.

While both Rate Limiting and TRW-CB use outgoing connections as the key detection feature, Figure 2 shows that Rate Limiting is less sensitive to p2p traffic as compared to TRW-CB. We noticed that the low sensitivity of Rate Limiting is because it operates on a long-term profile of traffic by keeping new connections in a queue. P2P applications establish a large number of connections, but in a short span of time during bootstrap. Therefore, Rate Limiting's queue threshold was not exceeded during this short-term connection activity period. On the other hand, the affect of p2p bootstrapping becomes very pronounced for TRW-CB which keeps changing its score with each individual connection attempt. Despite the low degradation observed in Rate Limiting, we note that the Rate Limiting detector generally provides the worst accuracy among all the evaluated detectors. Therefore, while its relative accuracy degradation in the presence of p2p traffic is low, its overall accuracy is considerably lower than TRW-CB; at 20% false positive rate, TRW-CB gives approximately 26% better detection rate than Rate Limiting. Hence, TRW-CB, despite having a larger accuracy degradation, should still be the preferred choice of portscan anomaly detector.

The accuracy degradation observed for Maximum Entropy is due to its reliance on a baseline distribution of destination port numbers. P2P peers generally use random port numbers which result in a distribution approaching uniformity which is incorrectly classified as malicious by the Maximum Entropy detector.

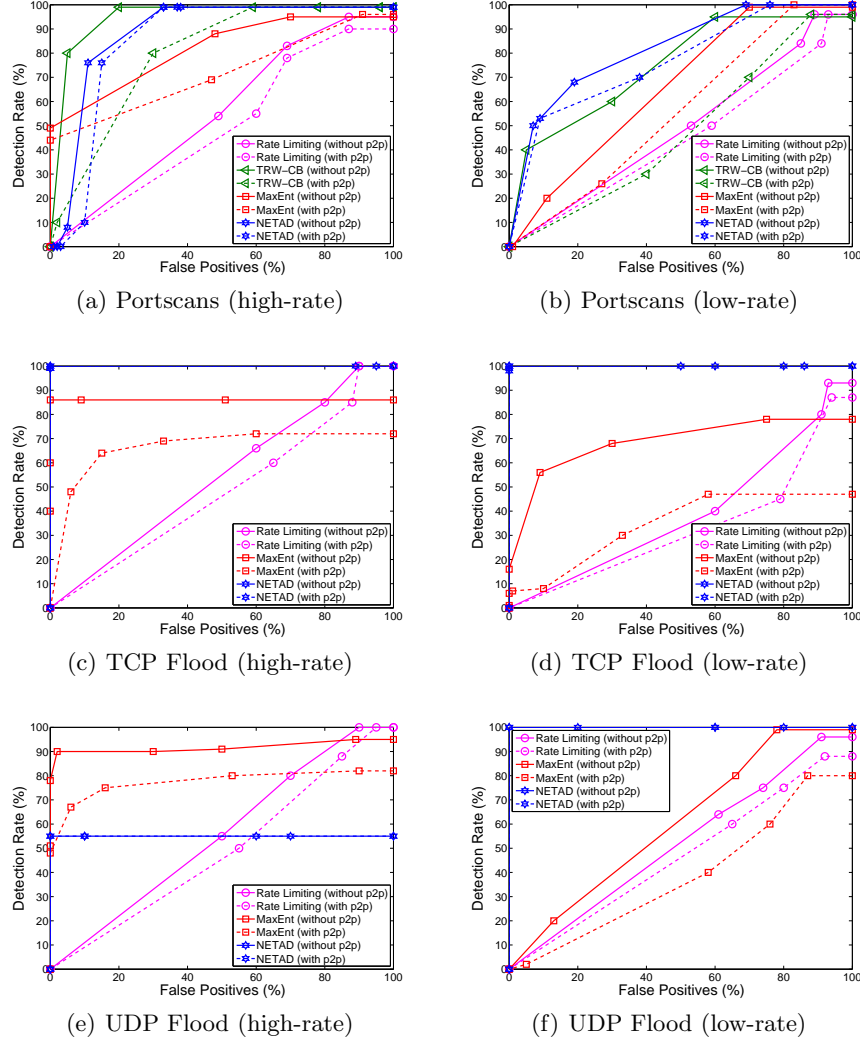


Fig. 3. ROCs for different attack classes/rates; results of TRW-CB for flooding attacks are omitted as it had 0% detection rate due to random source IP address spoofing used by flooding attacks.

4.3 Does the aggressive nature of p2p traffic dominate some/all attack classes and high-/low-rate attacks?

We now move to the question about whether or not p2p traffic has the same impact on different attack classes and rates. To address this question, Figure 3 plots separate ROCs for each attack class and rate. We note that the performance of NETAD does not degrade for flooding attacks when p2p traffic is

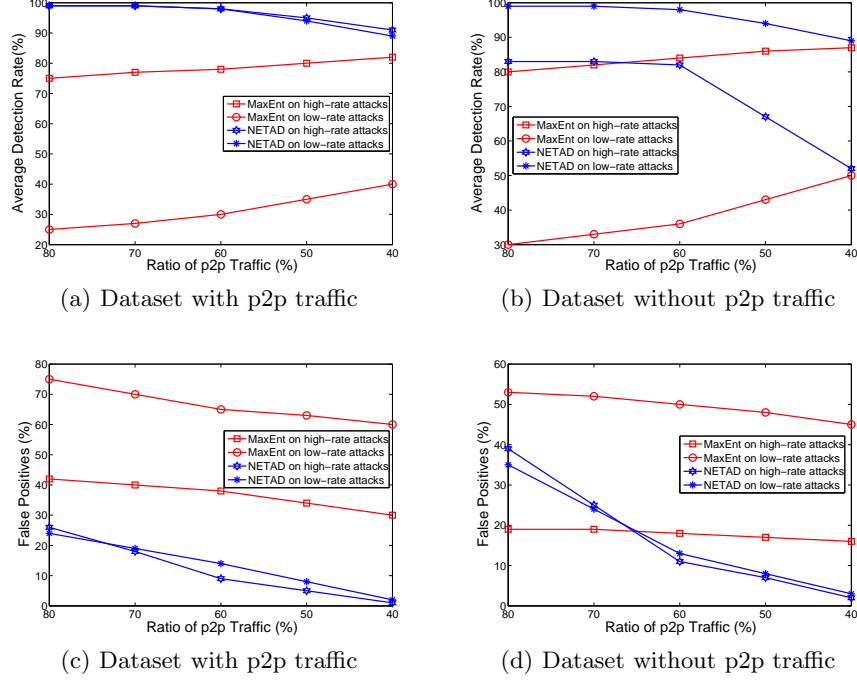


Fig. 4. Results of training IDSs on p2p traffic.

introduced, but its accuracy degrades for portscans. On the other hand, performance penalty for Maximum Entropy in case of flooding attacks is much more than that for portscans. This is mainly because of the differing design principles of these anomaly detectors. Flooding attacks are detected by NETAD because the floods were launched on lower ports [Table 2], whereas p2p communication was using higher ports for communication. The same attacks degrade Maximum Entropy detector's accuracy because p2p traffic on higher ports increases the variance and entropy of the port distribution, thereby resulting in a large number of false positives from windows containing p2p activity. These results indicate that depending on the detection principles and features employed by an anomaly detector, the affect of p2p traffic can be much more pronounced for some attack classes.

We are also interested in determining how p2p traffic affects low- and high-rate attacks. From Figure 3, we observe that detection of low-rate attacks is much more seriously affected than that of high-rate attacks. Thus p2p traffic inadvertently acts as a very effective evasion cover for low-rate attacks. This evasion cover is also provided for high-rate attacks, but the cover is blown for the scenarios where the sustained attack connection rate overwhelms the short-term p2p connection burst.

4.4 Can an anomaly detector handle p2p traffic if it is trained on a dataset containing p2p traffic?

Our performance evaluations thus far have indicated that the p2p traffic adversely affects the accuracies of *all* anomaly detectors evaluated in this work. We now investigate whether training a detector on p2p traffic can mitigate this torrent effect. To this end, we develop training sets with a proportion of p2p traffic which has been reported in Internet study reports [1]. We vary the proportion of p2p traffic in the training set from 40-80% and train NETAD and Maximum Entropy on this training set; TRW-CB and Rate Limiting do not require training and therefore we do not need to evaluate them in the present context. We then evaluate accuracies of the anomaly detectors on the entire dataset (containing all the p2p, malicious and background traffic).

Figure 4 shows the results for training NETAD and Maximum Entropy on different proportions of p2p traffic. It can be clearly seen from Figure 4 that training Maximum Entropy on p2p traffic not only degrades its accuracy but also increases its false positives rate. In case of NETAD, although we observe an increase in detection rate, a 30% increase in false positive rate is induced as we increase the amount of p2p traffic in the training set. This is mainly because p2p clients communicate with each peer on different ports and therefore it is not possible to define an effective filtering rule for NETAD or derive a robust baseline distribution for Maximum Entropy. Hence we conclude that training these anomaly detectors on p2p traffic does not mitigate the torrent effect mainly because contemporary detectors are not designed to filter or incorporate the peculiarities of p2p protocols and clients.

5 Mitigating the Torrent Effect

Based on the empirical accuracy results of the last section, in this section we discuss how can we make an anomaly detector resilient to p2p traffic. While the *right* method to make an anomaly detector resilient to p2p traffic is to avoid detection features which overlap between malicious and p2p traffic, in this section we only discuss an ad hoc method that can be used to make existing IDSs work with p2p traffic. In the following section, we will discuss how future anomaly detectors can inherently cater for p2p traffic in their design philosophy.

5.1 Can a pragmatic solution be designed to make an anomaly detector insensitive to p2p traffic?

Our evaluations in Section 4 show that the torrent effect is mainly caused by initiation of a large number of connections by p2p applications and failed connection attempts in those connections. This behavior of p2p applications is a result of: 1) lack of a central repository in p2p networks to maintain up-to-date information of available peers; and 2) ensuring robustness in p2p networks even with high churn rates. While these key design features of p2p networks can be

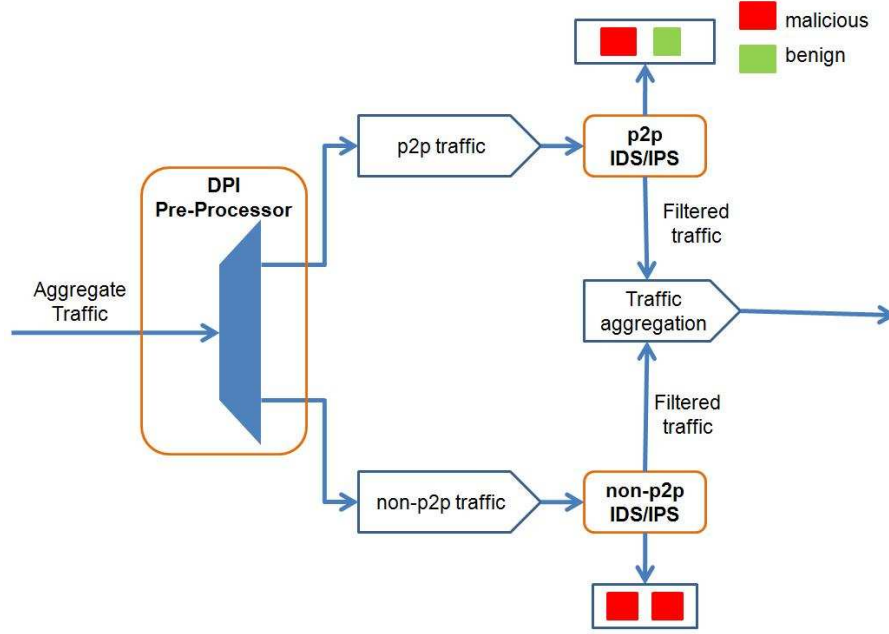


Fig. 5. Mitigating the torrent effect: An IDS with a p2p traffic classification based pre-processor.

achieved in a less aggressive manner, p2p applications perform unrelenting attempts to establish connections to thwart techniques to curb p2p connections. The means used to achieve these design goals of p2p networks result in an overlap with malicious behavior.

Since p2p protocols are unlikely to change their behavior in the near-term, and as an IDS designer cannot assume any control over these applications' behaviors, a simple solution to mitigate the torrent effect is to filter p2p traffic at the input of an anomaly detector using a p2p traffic classifier. Filtering of p2p traffic will result in segregation of non-p2p and p2p traffic as shown in Figure 5. Such a pre-processing filter can be followed by the IDS logic which, in the present context, will only operate on the non-p2p traffic; anomaly detection on the segregated p2p traffic will be discussed in the following section. Since contemporary IDSs are designed to work with non-p2p traffic, detection in the segregated non-p2p traffic will be performed on the unique and non-overlapping characteristics of malicious traffic, thereby yielding good accuracies. This p2p traffic classification based solution has an additional advantage that it requires no changes to be made to existing IDSs. Consequently, at the cost of higher complexity, this generic p2p traffic classification based pre-processor can be integrated into *any* anomaly detector.

Table 3. Mitigating P2P Effect Using P2P Traffic Classifiers Based Traffic Filtering (DR= Detection Rate; FP= False Positive; KPC= Karagiannis’ Payload Classifier)

| | Rate Limiting | | TRW-CB | | MaxEnt | | NETAD | |
|---------------------|---------------|-----|--------|-----|--------|-----|-------|-----|
| | DR% | FP% | DR% | FP% | DR% | FP% | DR% | FP% |
| No filtering | 50 | 45 | 60 | 22 | 62 | 48 | 65 | 25 |
| OpenDPI[23] | 56 | 43 | 64 | 12 | 63 | 32 | 70 | 17 |
| KPC[24] | 60 | 40 | 70 | 6 | 66 | 17 | 77 | 13 |

Table 4. Evaluation of OpenDPI and KPC on Encrypted P2P Traffic

| | Classified as p2p | Classified as unknown | Classified as non-p2p |
|----------------|-------------------|-----------------------|-----------------------|
| OpenDPI | 3.8% | 96.2% | 0% |
| KPC | 64.7% | 35.2% | 0% |

There are two problems with this p2p traffic filtering solution: 1) An IDS’ accuracy in this design is closely tied to the accuracy of the p2p traffic classifier, i.e., if the p2p traffic classifier can classify p2p traffic accurately, then anomaly detection accuracy will improve, and vice versa; 2) Attacks embedded within p2p traffic will not be detected. The rest of this section address the first point, while the second point is deferred to the next section. In particular, the next subsection answers the following question: Can existing public p2p traffic classification solutions mitigate the torrent effect?

5.2 Can existing public p2p traffic classifiers mitigate the torrent effect?

The p2p traffic classification problem has been well investigated and signature- and heuristic-based solutions exist. We, however, argue that many existing heuristic-based solutions will also be subject to the overlapping feature limitation.⁶ Therefore, it is important to choose approaches which use non-overlapping heuristics. We now evaluate our proposed design on a popular DPI-based technology and on a hybrid scheme (signatures + heuristics).

We perform traffic filtering using OpenDPI [23] (a signature based solution with over 90 signatures) and Karagiannis’ Payload Classifier(KPC) [24] (a hybrid solution with over 59 signatures); we refer interested readers to the original papers for the details of OpenDPI and KPC. The results of evaluation of the four anomaly detectors on filtered traffic are shown in Table 3. Table 3 shows that KPC (unknown: 35.2%) provides remarkably better accuracy than OpenDPI (unknown: 96.2%), mainly because OpenDPI is unable to detect any encrypted p2p traffic. It can be clearly seen by comparing Table 3 and Table 4 that the improvements in anomaly detectors’ accuracies are dependent on the traffic classifier’s accuracy. One of the limiting factors in the accuracy of the traffic classifiers is encrypted traffic.

⁶ For example, the method in [20] uses failed connections as a feature and should not be used in the present context.

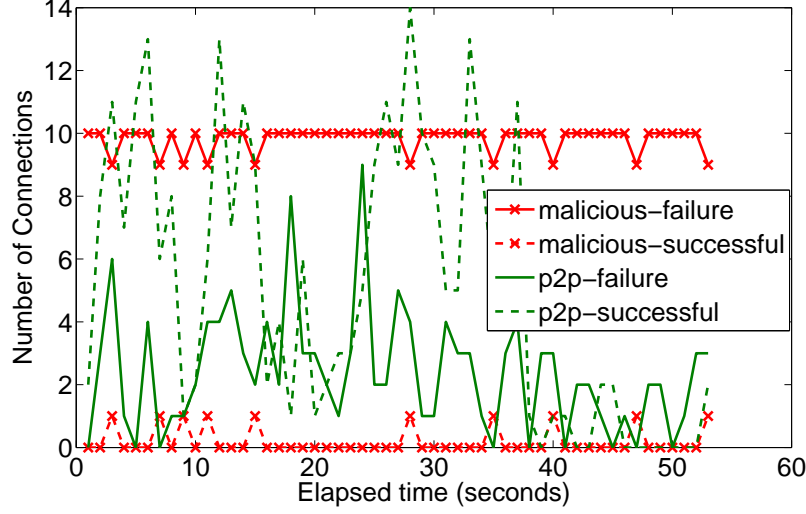


Fig. 6. Connection timeline for p2p and malicious (portscan attack) traffic.

We note from Table 3 that the current traffic classification accuracies of the DPI solutions are inadequate to induce a significant improvement in anomaly detection accuracy; detection rates after p2p traffic classification range from 40-70%, while false positives are between 6-40% for different anomaly detectors. Since the accuracies reported in Table 3 are impractical for commercial deployments, we conclude that public p2p traffic classification solutions at present cannot provide acceptable accuracies to induce an effective accuracy improvement in anomaly detection. While many commercial p2p traffic classification solutions are available, to the best of our knowledge, none of the p2p traffic classifiers proposed by the research community have acceptable detection accuracies for encrypted p2p traffic. Therefore, efficient p2p traffic classification remains an open problem and a solution to this problem will benefit the IDS community as well as the traffic engineering community.

Until such a solution is developed, we need to identify non-overlapping (between malicious and p2p) traffic features that an anomaly detector can rely on. As a preliminary, result, Figure 6 shows the connection timeline for the p2p and malicious traffic. It can be seen that the sustained activity of maliciousness is very different from the sporadic p2p traffic activity. Therefore, p2p and malicious traffic can be isolated if a notion of long-term statistics can be introduced during anomaly detection. This is part of our ongoing research.

6 What are the Open Problems in Designing Future Anomaly Detectors?

The tremendous growth in p2p-based file sharing, VOIP and video streaming traffic has revolutionized the Internet traffic characteristics. Our evaluations showed that this change in traffic characteristics cannot be characterized by existing anomaly detectors which rely on traffic features (e.g., rate, connection failures, ports, etc.) that largely overlap with p2p traffic behavior. While we proposed an adhoc solution which allows existing IDSs to work effectively, a question remains open regarding the scalability of this solution to future Internet traffic. Recent projections of future attacks show that some of the greatest threats in the future will be originating from file sharing networks [28]. In such a threat landscape, a p2p traffic classification based solution will simply allow all malicious activities embedded within p2p traffic to go undetected.

While detection of malware delivered using p2p applications does not fall under the scope of traffic anomaly detection, attacks originating from p2p networks should be detected using these IDSs. One such attacks has already been proposed in [27] where Naoumov and Ross designed a DDoS engine for flooding a target using the indexing and routing layers in a p2p systems. Similarly, IDSs should be able to detect the exploits targeted at vulnerabilities which are a product of the change to firewall rules for p2p traffic [29]. Finally, it is highly desirable to detect the C&C channels of bots which also use p2p communication [30].

Given the premise that p2p traffic is here to stay, our work demonstrates the need to rethink the classical anomaly detection design philosophy with a focus on performing anomaly detection in the presence p2p traffic. We argue that p2p traffic classification will play a fundamental role in future IDSs as it will facilitate detection of both the p2p and the non-p2p traffic anomalies, as shown in Figure 5. In our proposed design, traditional non-p2p network attacks will be detected using existing anomaly detectors, while an additional IDS that specializes at detecting attacks within p2p traffic will also be deployed.

Design of a p2p-specialized IDS is still an open research problem that is part of our ongoing research and that we also expect our peers to follow-up on. We have made our dataset publicly available for performance benchmarking of such future IDSs and p2p traffic classifiers.

Acknowledgments. We thank Dr. Hyun-chul Kim for providing Karagiannis' Payload Classifier.

References

1. Ipoque Internet Study Report 2008/2009, http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009
2. Maier G., Feldmann A., Paxson, V., Allman, M.: On Dominant Characteristics of Residential Broadband Internet Traffic. In IMC (2009)

3. Erman, J., Gerber, A., Hajiaghayi, M.T., Pei, D., Spatscheck, O.: Network-Aware Forward Caching. In: WWW (2009)
4. Labovitz, C., McPherson, D., Iekel-Johnson, S.: 2009 Internet Observatory Report. In: NANGO: NANGO47 (2009)
5. Li, Z., Goyal, A., Chen, Y., Kuzmanovic, A.: Measurement and Diagnosis of Address Misconfigured P2P Traffic. In: IEEE INFOCOM (2010)
6. Jung, J., Paxson, V., Berger, A.W., Balakrishnan, H.: Fast Portscan Detection Using Sequential Hypothesis Testing. In: IEEE Symposium on Security and Privacy (2004)
7. Schechter, S.E., Jung, J., Berger, W.: Fast Detection of Scanning Worm Infections. In: RAID (2004)
8. Williamson M.M.: Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code. In: ACSAC (2002)
9. Twycross J. Williamson, M.M.: Implementing and Testing a Virus Throttle. In: Usenix Security (2003)
10. Gu, Y., McCullum, A., Towsley, D.: Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation. In: ACM IMC (2005)
11. Mahoney, M.V.: Network Traffic Anomaly Detection Based on Packet Bytes. In: ACM Symposium on Applied Computing (2003)
12. Next-Generation Intrusion Detection Expert System (NIDES), <http://www.csl.sri.com/projects/nides/>
13. Weaver, N., Staniford, S., Paxson, V.: Very Fast Containment of Scanning Worms. In: Usenix Security (2004)
14. Lakhina, A., Crovella, M., Diot, C.: Diagnosing Network-wide Traffic Anomalies. In: ACM SIGCOMM (2004)
15. Lakhina, A., Crovella, M., Diot, C.: Mining Anomalies Using Traffic Feature Distributions. In: ACM SIGCOMM (2005)
16. Patcha, A., Park, J.: An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Elsevier Computer Networks (2007)
17. DARPA Intrusion Detection Data Sets, <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html>
18. LBNL/ICSI Enterprise Tracing Project, <http://www.icir.org/enterprise-tracing/download.html>
19. Endpoint Dataset, <http://wisnet.seecs.edu.pk/projects/ENS/DataSets.html>
20. Collins, M., Reiter, M.: Finding Peer-to-Peer File-Sharing Using Coarse Network Behaviors. In: Proceedings of the European Symposium On Research In Computer Security (2006)
21. Bartlett, G., Heidemann, J., Papadopoulos, C.: Inherent Behaviors for On-line Detection of Peer-to-Peer File Sharing. In: Proceedings of the 10th IEEE Global Internet (2007)
22. Liu, Y., Guo, Y., Liang, C.: A Survey on Peer-to-Peer Video Streaming Systems. In: Peer-to-peer Networking and Applications (2008)
23. OpenDPI, Ipoque's DPI software's Open Source Version, <http://www.opendpi.org/>
24. Karagiannis, T., Broido, A., Brownlee, N., Claffy, K.C., Faloutsos, M.: Is P2P Dying or Just Hiding? In: IEEE Globecom (2004)
25. Sun, X., Torres, R., Rao, S.: DDoS Attacks by Subverting Membership Management in P2P Systems. In: 3rd IEEE Workshop on Secure Network Protocols (2007)
26. Athanasopoulos, E., Anagnostakis, K.G., Markatos, E.P.: Misusing Unstructured P2P Systems to Perform DoS Attacks: The Network That Never Forgets. In: ACNS (2006)

27. Naoumov, N., Ross, K.: Exploiting P2P Systems for DDoS Attacks. In: INFOSCALE (2006)
28. 2010 Cyberthreat Forecast from Kaspersky Lab, http://usa.kaspersky.com/about-us/news-press-releases.php?smnr_id=900000322
29. Chien, E.: Malicious Threats of Peer-to-Peer Networking. Whitepaper, Symantec Security Response (2008)
30. McAfee Labs 2010 Threat Predictions, http://www.mcafee.com/us/local_content/white_papers/7985rpt_labs_threat_predict_1209_v2.pdf
31. Arbor Peakflow: IP Traffic Flow Monitoring System, http://www.arbornetworks.com/index.php?option=com_content&task=view&id=1465&Itemid=692
32. Allot Service Protector, DDoS Protection, http://www.allot.com/Service_Protector.html#products
33. Sandvine: Network Protection, http://www.sandvine.com/products/network_protection.asp
34. Ipoque Press Release: P2P Raid in Germany Shows Little Effect, <http://www.ipoque.com/news-and-events/news/pressemitteilung-ipoque-210606.html>
35. Ashfaq, A.B., Robert, M.J., Mumtaz, A., Ali, M.Q., Sajjad, A., Khayam, S.A.: A Comparative Analysis of Anomaly Detectors under Portscan Attacks. In: RAID (2008)
36. Javed, M., Ashfaq, A.B., Shafiq, M.Z., Khayam, S.A.: On the Inefficient Use of Entropy for Anomaly Detection. In: RAID (2009)