Resistance is Not Futile: Detecting DDoS Attacks without Packet Inspection

Arjun P. Athreya, Xiao Wang, Yu Seung Kim, Yuan Tian and Patrick Tague

Wireless Network and System Security Group,
Carnegie Mellon University, USA.
{arjuna,xiaowang,yuseungk,yt,tague}@cmu.edu

Abstract. Packets in anonymous networks are fully protected. Therefore, traditional methods relying on packet header and higher layer information do not work to detect Distributed-Denial-of-Service (DDoS) attacks in anonymous networks. In this paper we propose to use observable statistics at routers that need no packet inspection to infer the presence of an attack. We propose packet resistance as a metric to detect the presence of attacks which reduce the availability of channel bandwidth for wireless routers in the core network. Our proposed detection framework is distributed, wherein each router in the network core monitors and reports its findings to an intermediate router. These intermediate routers form a hierarchical overlay to eventually reach a centralized attack monitoring center. The alarm messages are used to construct an attack path and determine the origin of the attack. We present simulation results to demonstrate the effectiveness of our proposed metric.

1 Introduction

In many communication applications (email, social networking, peer-to-peer file sharing, sensitive battlefield military communications for example) the end users share sensitive data between them. The data and communication patterns learned from networks sharing such sensitive information leads to privacy and other security breaches [1]. Hence, for reasons such as confidentiality, political restraint evasion or sensitive communications, network anonymity is desired.

Network anonymity protects against traceability of end-hosts of networks even though their data traverses through the network in the presence of adversaries. In a fully anonymized network using services such as Tor [2], all the packet's contents and the packet's meta data are fully protected by cryptographic techniques. These packets offer no flow information, source and destination IP addresses, other IP header details or even higher layer details of the applications being supported. Therefore if attackers inject or flood such networks with garbled or replayed packets, the routers cannot distinguish between flows of legitimate traffic versus attack traffic. Thus traditional attack detection mechanisms do not work for anonymous networks. This is because traditional network attack detectors monitor network parameters and meta-data of network packets (IP addresses, packet sequence numbers, packet sizes) and then looks for their anomalies in their behavior to raise attack alarms [3] [4] [5]. Many of these attack detectors are designed to raise alarms in nearly real-time [6]. Cryptanalysis is one way of breaking cryptographic properties of these fully encrypted messages, but such mechanisms do not always yield plain text information in real time.

Anonymous networks themselves offer no insights of data packet traffic flowing through them. However routers in such networks could offer statistics which are easily observable. Statistics such as packet count, packet dropping rate at a router's interface can be observed without needing packet inspection. Such statistics could reveal the current performance of the attack detection metrics and observe anomalies to detect the presence of an attack. Given that these statistics are observable with no packet inspection needed, attack detectors using such statistics could raise attack alarms in near real-time. Studies and research efforts are still being pursued to de-anonymize users of anonymous networks [7] [8] [9], but to our best knowledge designing a real-time DDoS detector for anonymous networks remains to be a hard and unsolved problem.

In this paper we propose a statistical distributed and hierarchical attack detector for anonymous networks using observable statistics at routers in the core network. In this work, our attack model focuses on DDoS attacks which reduce available channel bandwidth to routers in the core network. The attacks either flood the network with data packets or exhaust bandwidth by misbehaving with the medium access control protocols [10].

Our contribution for the proposed statistical DDoS detection is as follows,

- *Packet Resistance* metric for local monitoring: We define packet resistance as the ratio of incoming (receive packet rate) data rate over outgoing (transmit packet rate) data rate. This metric therefore needs no packet inspection.
- Local Analysis on Packet Resistance: We monitor the proposed metric to detect attacks and raise alarms. Packet resistance increases when a router experiences opposition to outgoing traffic on its egress interface. This indicates the presence of an attack.
- Hierarchical decision aggregation: We allow for locally generated alarms to be collected by an overlay of intermediate routers and eventually reach a centralized attack monitoring center.
- We use aggregated alarm data to construct an attack path to determine the possible origin of the attack.
- We simulate our work using the CORE simulator for a core network of 50 nodes and present our detection accuracy using the packet resistance metric.

The remainder of the paper is organized as follows. In Section 2 we discuss our system assumptions and network model. We propose our statistical attack detection framework in Section 3. We discuss local attack monitoring in Section 4. Then we study the use of local attack reports to construct attack paths and determine the location of the intruder in Section 5. Finally we discuss conclusion and our future work in Section 6.

2 Network and Attack Model

We discuss our network and attack model for our work in this section. We describe a network architecture for our work that illustrates the anonymous communications between users of trusted groups.

2.1 Network Model

Our network model comprises end-hosts in various trusted groups communicating with each other via a network core as shown in Fig. 1. The network core comprises several wireless local area network (WLAN) using the IEEE 802.11 communications standard. The routers of a WLAN share the channel bandwidth supported by the wireless communication standard followed in the WLAN. These routers have multiple interfaces and therefore can act as gateways to two or more WLANs. The packets leaving the trusted groups are anonymized by the group's gateway. The core network in this work is a fully wireless network deployed by trusted groups or trusted organizations (for example: military or private corporations). We assume that the hardware and software on these routers are tamper-proof and trustworthy. Each of the routers deployed have attack detection sensors on them which can monitor and report the metric's performance.

2.2 Attack Model

Our attack model involves launching a Distributed-Denial-of-Service (DDoS) attack in the core network. The DDoS attack in our work affects channel bandwidth of the WLAN in the core network. We consider two ways in which this DoS can be launched. First, the target router is attacked in such a way that the net incoming rate on all its ingress interfaces exceeds its forwarding rate on the egress interface. For example, in the IEEE 802.11 standard the maximum data rate is 54 Mbps. If a router has two ingress interface receiving at 40 Mbps each and one egress interface, it can only send at a maximum data rate of 54 Mbps while its net incoming data rate is 80 Mbps. We consider this as an instance of flooding the router to create the DoS on the egress link of the target router. Second, available bandwidth for a router is exhausted by through medium access misbehavior [10]. Thus both these attacks render the target router to lower its forwarding rate in the WLAN.

Our attack model is distributed in the sense that the flooding could be initiated by multiple attackers located distributed in the core network. Specific links could be targeted by attackers distributed in the network as demonstrated in the Coremelt Attack [11]. Hence this leads to a DDoS attack framework.

We consider three attack traffic models: bursty, periodic and random. The bursty attack floods the victim (target) router occasionally such that there is a sudden surge in data traffic at the router. For bursty traffic, we allow the attacker to periodically flood the network with high volume traffic but with very short duration and then send no traffic during other times. We set the duration of the pulse to 5 seconds. The periodic attack exhibits the same attack pattern as

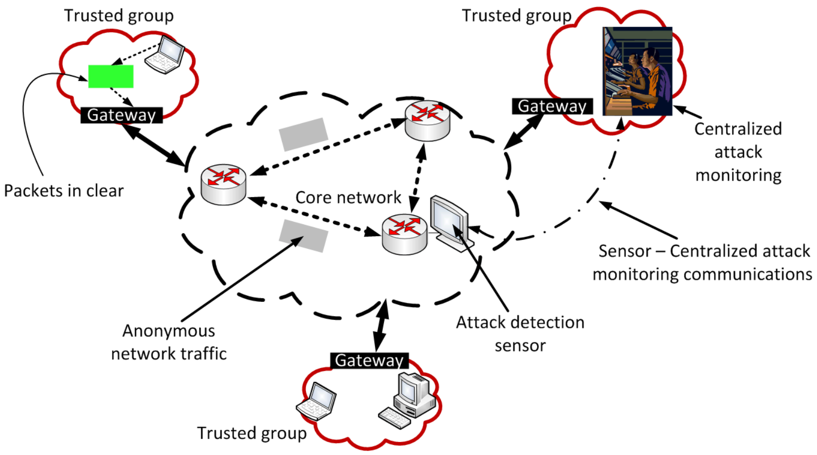


Fig. 1: We illustrate anonymous communication in the core network supporting communications between end-hosts in trusted groups. The attack detection sensors in the core report to centralized attack monitoring centers in the trusted groups.

bursty attack, but does it periodically. Finally, the random attack injects attack traffic at random instances of time. The volume of attack traffic and the length of attack can be varied.

3 DDoS Detection in Anonymous Wireless Networks

We take a top-down approach to describe our attack detection framework for anonymous networks. Our framework has two primary components: local analysis of network performance statistics at each router and aggregation of alerts or decisions using hierarchy. In what follows, we describe these two components and the associated challenges.

3.1 Local Analysis of Router Statistics

In anonymous networks, the packets reveal nothing about the services generating these packets nor the source and destination hosts in trusted groups. This makes it difficult to distinguish the attack traffic packets from legitimate network traffic packets in real-time which otherwise would have helped in detecting an attack at such routers. In order to determine the presence of attacks which affect the channel bandwidth, we observe the aggregate packet statistics such as received packet counts, transmitted packet counts at each router over periodic time intervals that help illustrate the trend of the forwarding rate at the respective router. If a router experiences drops in its forwarding rate while its incoming rate is still the same, then it raises an alarm. At this instance the router raises an attack alarm which conveys the interface it is experiencing the attack. We allow each

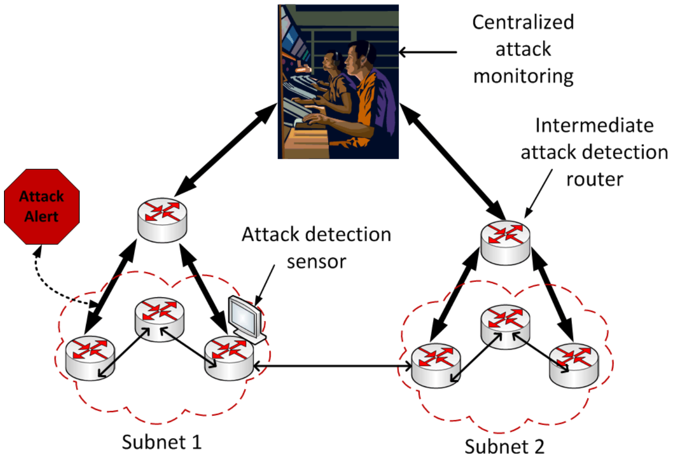

Centralized attack monitoring
Intermediate attack detection router
Attack Alert
Attack detection sensor
Subnet 1
Subnet 2

Fig. 2: We illustrate an example of scalable hierarchical attack detection system. Intermediate attack detection sensors collect alarm messages from a group of routers and report the alarms to a central attack monitoring center.

router to raise an alarm when it experiences such a phenomenon and thereby multiple alarm messages could be used to understand the attack's effect in the core network.

3.2 Hierarchical Decision Aggregation

Practical deployments of anonymous networks comprise a large number of routers [12]. This makes all routers directly reporting to one centralized attack detection monitoring center not feasible. A hierarchical attack detection model is thus desired to allow for scalability and improving accuracy. It is possible when local monitoring reports be outliers which are errors in detecting the attack. However, when reports from several routers are aggregated, the error in falsely detecting attacks can be reduced by observing the reports from other adjacent routers. In a hierarchical attack detection model, a set of routers report the alarms to a gateway router of a subnet of routers. These routers could make local decisions at a subnet level or forward all the alarms to its supervising attack detection node which could be monitoring several other subnets. With this, a hierarchy of attack monitoring routers could create a logical overlay in a larger network to eventually report to a central attack detection center as shown in Fig. 2 .

3.3 Challenges with Aggregate Statistics

The attack detection metrics based on observable and aggregated statistics while being non-invasive also pose challenges in attack detection in anonymous networks. Aggregated statistics are easy to collect at routers and statistical operations such as deviations or variances do indicate the changes in metric with time. Particularly for wireless networks, operations on aggregated statistics do not always provide straight forward intuitions of attack's existence. Bandwidth consumption is not constant in wireless networks even when there are no attacks. The stochastic nature of the wireless channel imposes large variations in the metric. The stochastic nature of wireless channels could result from factors such as changes in propagation environment, mobility and different propagation models. Hence, the variance of metrics about its mean will not be small as could be seen in wired networks which can capture certain instances of flooding attacks with relatively good accuracy [11] [13]. Metric variation could also result due to higher layer performance optimization such as TCP. Traffic could surge until TCP starts to back off and during this brief period of time, it is possible that the metric shoots beyond the ideal value leading to an alarm being raised. This leads to the fact that false alarms could be raised by attack detectors in anonymous networks which rely on raw aggregated statistics for attack detection.

4 Local Attack Monitoring using Packet Resistance

To demonstrate the use of our statistical DDoS detection framework, we propose a specific router statistic that can easily observed in anonymous networks, yet promising detection capabilities. In what follows, we propose the *packet resistance* metric, discuss its practical aspects and provide an experimental study.

4.1 Packet resistance

To determine the increase in opposition to outgoing traffic at a router, there are two ways to observe this phenomenon at routers. One is to observe the rate at which incoming packets arrive and monitor the packet queue at routers. This however is tied to the fact that different routers could have different upper bounds on the queue lengths and thereby setting global thresholds to detect the presence of an attacker is not possible. However, we can monitor both the incoming and outgoing packet rates at a router. Thus we can detect an attack if a router sends out packets at a rate lower than it receives.

In device physics, *resistance* of a conductor is defined as the opposition to the passage of electric current through it. In our work we define *packet resistance* of a router as the opposition to its packet transmission. In a sampling interval of t_s seconds at router i, reception of rx_i packets and subsequent forwarding of f_i packets can be interpreted as a *packet resistance* (PR) of $R_i = \frac{rx_i}{f_i}$, which is a unitless measure.

The PR metric measures the resistance faced by a router when the outgoing link is being attacked. For example, if the router is able to transmit packets nearly at the rate it receives them, then the average value of the metric should be 1.0 with very little variance. When the incoming packet rate exceeds the outgoing packet rate during the sampling interval, then the metric overshoots 1.0. While higher layer services such as TCP changes the rate at which packets are delivered to the network layer, the attackers may not follow such traffic control techniques and sustain their traffic which our metric can capture.

Other advantages of packet resistance metric are multiple fold. First, the metric is based on commonly available and easily observable statistics at routers. It needs no inspection of packets and makes no distinction between various flows of the data traffic, even including the attack traffic. The alarm is thus based on aggregate statistics observed at a router. Second, the metric needing no packet inspection means that the metric can be computed in near real-time and presents very minimal computation overhead at the router. Finally, since a wireless medium is shared by network routers, affecting one link could affect multiple other links in the network, thus there will be variations in the metric's performance on other network routers.

Metric Smoothing: To mitigate false alarms resulting from bursty and stochastic behavior of network traffic or traffic control mechanisms of higher layer services such as TCP, we propose a smoothing technique for the packet resistance metric. The smoothing technique is implemented in each router and takes only the PR metric's history as its input. The output will be used to help decide if the observed fluctuations in the metric is an indication of an attack or a genuine burst in the traffic.

The traffic at an interface of a router is sampled in intervals of t_s seconds and computes the metric for this time period. The router then maintains a historic average of the metric that gets reset at time intervals very large compared to t_s. This refresh interval for the historic average can be decided by network administrators based on traffic behavior over time. This historic average of the metric is our notion of measurement history on the interface of a router, assuming that it is serving the same network and connected to the same set of adjacent routers. This historic average of the metric at a router i is defined as $\overline{R_i}$.

If the router sounds an alarm every time the metric during the t_s interval overshoots a threshold T_r, the metric flags the measurement instance as an instance of flooding attack. If this is due to a genuine burst in traffic from an end-host, then we are capturing these bursts as *false positive* alarms. In order to mitigate the occurrences of false positives, we introduce a smoothing function for the proposed metric. While the interface samples the traffic at t_s seconds, we allow the router to observe the traffic for a time period of t_o seconds which we call as the observation interval. For simplicity purposes, we allow the t_o to be integer multiples of t_s. The t_o is a moving window which progresses after every t_s interval by 1 second. We define a function which has two input parameters, first, the deviation of the samples computed with respect to the historic average and second, the t_o. Let $t_o = k \cdot t_s$ and the value of the metric in k^{th} interval can be $R_i^{(k)}$. The deviation is computed as

$$D = \frac{\sqrt{\sum_{i=1}^{k} (R_i^{(k)} - \bar{R}_i)^2}}{k}. \qquad (1)$$

During t_o, if the metric average $\overline{R_i^{(k)}}$ during t_o exceeds a threshold T_r and its deviation D exceeds a threshold T_o, then we treat this as a possibility of an attack at the router.

However, the smoothing interval and the thresholds can be learned by the attacker over time. Hence, the attacker could get away by not flooding the link for the duration which will not be seen as a genuine burst in traffic. Such an instance is an example of a *false negative* alarm. In order to mitigate the instances of false negatives in our detection, we define t_d in seconds as the attack detection interval. We let t_d be a time window being integer multiples of t_o which progresses after every t_s seconds. During this larger time interval, we can count the number of instances ($Count$) where $\overline{R_i^{(k)}}$ and D overshot the thresholds T_r and T_o respectively during the t_o intervals within t_d. If PR and D overshot their respective thresholds during majority of the observation intervals within a t_d, then the router will raise an alarm to indicate that an attack is detected as shown in Fig. 3. We set T_d as two-thirds the number of t_o in t_d to indicate a majority.

```
Statistical Attack Detection
$\bar{R}_i \leftarrow 0$
while $R_i$
    for each $t_d$
        for each $t_o$
            $Count \leftarrow 0$
            for each $k \in t_s$
                $R_i^{(k)} \leftarrow \frac{rx_i}{f_i}$
            Update $\overline{R_i}$
            Compute $D$
            if $\overline{R^(k)_i} \geq T_r$ && $D \geq T_o$
               then $Count \leftarrow Count + 1$
        if $Count \geq T_d$
           then Raise Alarm
```

Fig. 3: We illustrate our statistical attack detection mechanism for detection DDoS in anonymous networks.

4.2 Experiment Setup

We implemented the attack methods and proposed detection techniques using the Common Open Research Emulator (CORE) [14]. Our network comprises 50 routers and use the OSPFv2 routing protocol to establish routes. The bandwidth for each link is 54 Mbps per the IEEE 802.11 standard. Since each router may have multiple radios, the total incoming traffic for a router can go above 54 Mbps with an upper bound of number of ingress interface times 54 Mbps. The routers were configured to report the attack alarms to a server managed by the trusted groups. For this setting we allowed for one server, while multiple such reporting servers could be setup for demonstrating a hierarchical detection framework. The attacker targets a router and introduces traffic flows at higher data rates to ensure the flooding is effective. We launched the attack traffic using $iperf$ and set the attack traffic to be UDP and of constant rate during each sampling interval [15]. Our sampling interval was set to 3 seconds, which was the lowest possible due to the constraints set by the simulator.

We chose various smoothing parameters to analyze the detection performance. We chose these parameters to vary the ratio between the evaluation interval and the detection interval. We selected 6 pairs of (t_o, t_d) parameter tuples for evaluation purpose: (1, 3), (3, 3), (3, 6), (6, 6), (10, 6) and (10, 10). Since we are constrained by space, we show the results observed at one router for the parameter tuples (1, 3), (6, 6) and (10, 10) as similar results were seen at other affected routers in the core network. The T_r was set to 1.0 and T_o was set to 0.1, 10% of T_r.

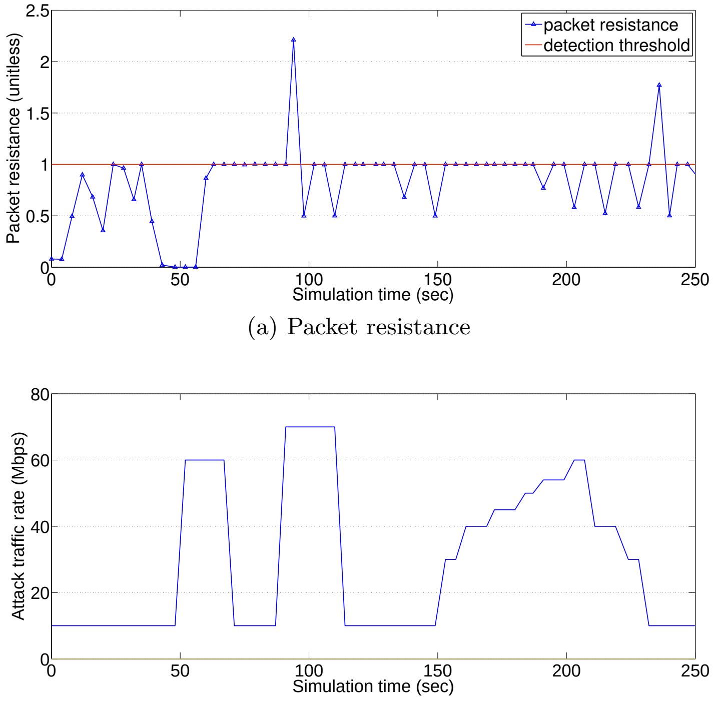
4.3 Results and Analysis

We evaluate the effectiveness of the proposed packet resistance metric to detect the presence of attackers who launch bandwidth consumption attacks.

In Figures 4 - 6 we illustrate the proposed metric for the three attack traffic models for the parameter tuple $(1, 3)$. In each of the Figures 4 - 6, sub-figure (a) illustrates the variation of the packet resistance metric and sub-figure (b) shows corresponding change in attacker traffic.

In Figures 4(a), 5(a) and 6(a), as the attack traffic floods a link of a target router, our initial observation was that the metric overshoots the threshold value of 1.0 as the outgoing traffic rate at the affected router is reduced. We further observe that there is a delay in decision making of about $1 - 2$ seconds due to processing of the observed data to make a decision at the end of decision interval. Our metric also captured the surge of outgoing traffic when the attacker traffic stops. This phenomenon is due to the fact that the router clears queued traffic at the router when its outgoing link is affected, which means the residue traffic has to be queued. Thus once the bandwidth frees up, the router makes an attempt to transmit the data at a rate higher than the incoming data rate, which leads to the metric value dropping below 1.0.

To analyze the detection performance, we use the false alarm rate, miss rate and accuracy. We first calculate the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) occurrences in our detection results for all parameter tuples and attack traffic models. True positive detection instances are those where the detector raised an alarm when there was actually an attack. True negative detection instances are those when the detection raised no alarm when there was no attack. False positive detection instances are those where



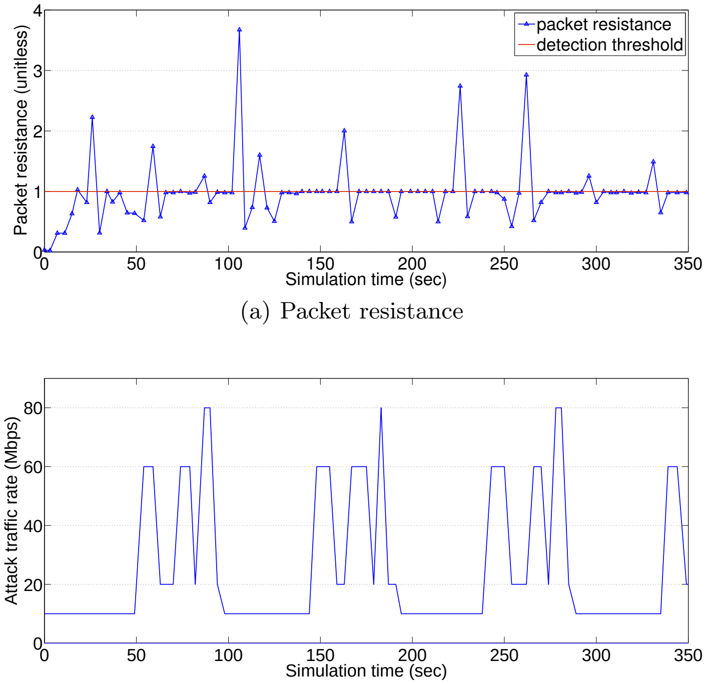
(a) Packet resistance

(b) Attacker traffic model: Random

Fig. 4: We demonstrate (a) the resulting packet resistance due to (b) random attack traffic with detection parameter (1,3). Our results suggest that the packet resistance metric can reflect the dynamics in attack traffic.

the detector raised an alarm when there was no attack. Finally, false negative detection instances are those when the detector raised no alarm when there was attack. False alarm rate refers to the ratio of false positve instances over instances when there are no attacks (false alarm rate = FP / (FP + TN)). Miss rate refers to the ratio of false negative instances over instances when there are actually attacks (miss rate = FN / (FN + TP)). Accuracy is given by the ratio of true detection instances over all instances (accuracy = (TP + TN)/(TP + TN + FP + FN)).

For all the three attack traffic models and the parameter tuples in Table 1, we observed that the attack detection accuracies by using a readily observable technique needed observation intervals longer than just one sampling interval. We see that the average prediction accuracy for all three attacker traffic models when observation interval t_o set to the sampling interval t_p is 75%. The false alarm rate is from 4% to 24%, depending on the attack traffic type. The miss rate hits 0 for tuple (6, 6) and (10, 10), but goes up to 96% for the tuple (1, 3).

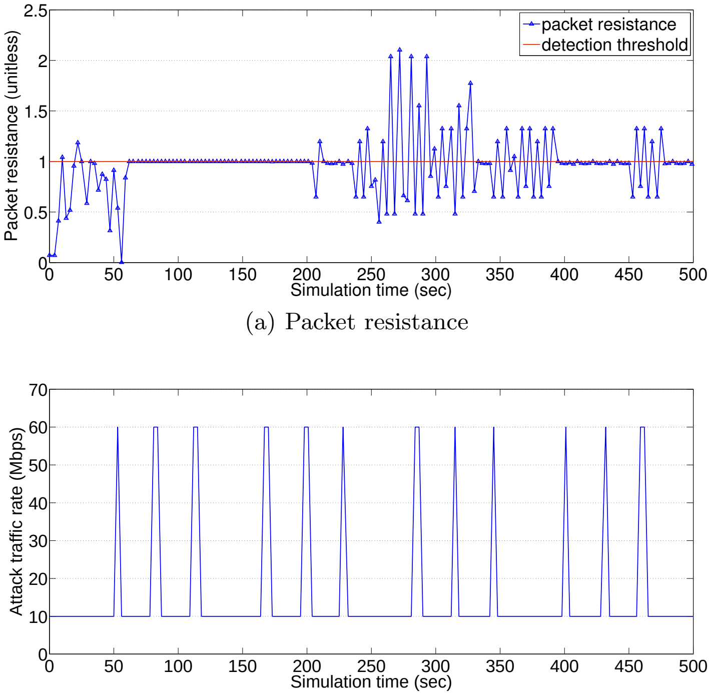


(a) Packet resistance

(b) Attacker traffic model: Periodic

Fig. 5: We demonstrate (a) the resulting packet resistance due to (b) periodic attack traffic with detection parameter (1,3). We observe that with short observation interval, the metric was able to detect the attack on 90% of the times when the periodic attacker traffic was in the network.

For other observation intervals greater than the sampling interval, the average detection accuracy is 98%. This shows that observing the traffic for duration longer than sampling interval allows us to capture the instant bursts in traffic either due to medium stochastic nature or due to the traffic bursty nature while higher layer protocols such as TCP converge. Additionally, for the detection interval which is several sampling intervals facilitates the monitoring of traffic for longer intervals of time. This allows for the natural bursts in traffic to normalize, while still being able to capture the presence of persistent attack traffic. We clearly observe that needing an observation interval greater than sampling interval improved detection accuracy for the periodic and bursty attack traffic. However for the random attack traffic, the tuple $(1, 3)$ is sensitive to traffic changes which results in high miss rate which should have yielded higher false positives instead of false negatives. But this is a trend which needs further investigation as this bias could be due to measurement error or lack of enough metric samples. While we observe these trends, it still remains to be an interest-



(a) Packet resistance

(b) Attacker traffic model: Bursty

Fig. 6: We demonstrate (a) the resulting packet resistance due to (b) bursty attack traffic with detection parameter (1,3). Though the attacker introduced short interval traffic bursts into the network and remained offline for remainder of the time, even with the observation interval equal to one sampling period, the metric captured the presence of attack with close to 90% accuracy.

ing question as to what is the optimal detection interval for a chosen observation interval to maximize the detection accuracy

From the detection results in Table 1 and the visual illustration of the metric behavior for various attack traffic models, we show that statistical attack detection using purely observable metrics yields good attack detection accuracy for anonymous networks. Additionally, these detections were made in near real-time needing a few seconds of delay for processing and decision making. Thus we need not break the cryptography to learn about the presence of intruders who are able to attack the core network bandwidth. We are not claiming this metric to always detect local attacks with such high detection accuracy since our simulation setup and data sample set is small. However, when local monitoring alarms are used by a hierarchical attack detection system with lower accuracy, then the attack detection accuracy of the for the whole system will be high due to aggregation of larger number of attack reports.

Table 1: We illustrate the detection accuracy for our proposed statistical attack detection framework using packet resistance metric. The detection accuracy are shown for 3 types of attack traffic models and various detection parameter tuples. The parameter tuple comprises the observation interval which is integer multiples of the sampling interval and the detection interval which is integer multiples of the observation interval.

Attacker traffic	Parameter	TP	TN	FP	FN	False alarm rate	Miss rate	Accuracy
Random	(1,3)	1	41	2	25	0.0465	0.9615	0.6087
	(6,6)	2	81	0	0	0.0000	0.0000	1.0000
	(10,10)	2	83	3	0	0.0349	0.0000	0.9656
Periodic	(1,3)	3	77	9	3	0.1047	0.5000	0.8696
	(6,6)	9	69	0	0	0.0000	0.0000	1.0000
	(10,10)	2	51	2	0	0.0377	0.0000	0.9636
Bursty	(1,3)	61	364	115	0	0.2401	0.0000	0.7870
	(6,6)	6	69	2	0	0.0282	0.0000	0.9740
	(10,10)	71	226	0	0	0.0000	0.0000	1.0000

5 Tracing the Attack's Origin

In Fig. 7, we illustrate the procedure to infer the source of flooding attacks in an anonymous network. When an attack is locally detected at each router, each router sends an alarm with the wireless network interface where the attack happens. In the wireless network we model, each router detects an attack when the amount of outgoing traffic exceeds the assigned bandwidth at each wireless interface. The receiving network interface, on the other hand, always receives the amount of traffic less than or equal to the assigned bandwidth. In Fig. 7, R_3, R_4, R_7, and R_8 report the flooding alarm with their respective detected network interface to the centralized attack detection center. In this example, the attack is detected at all l_2 interfaces of routers. The centralized attack detection center can make a decision on the source of flooding attack by reconstructing the attack path. Since the number of paths (between routers) via each wireless network interface is different, the centralized attack detection center can first include all the paths used by the reported network interface in the set of candidate attack paths. In this example, the set will have the path element R_3-R_5, R_3-R_4, R_4-R_7, R_7-R_6, R_7-R_8, and R_8-$Victim$. Then, the centralized attack detection center gets rid of the paths to the router that no alarm is reported (e.g., R_3-R_5 and R_7-R_6). Consequently, the attack path will be conjectured as R_3-R_4-R_7-R_8-$Victim$.

We note that one attack flow might not generate alarms on all other routers on a given path. But with each router locally monitoring the attack raises an alarm based on independently occurring attacks. As the routers have no global network knowledge, the extent of attack propagation will not be known to the individual routers. Hence, using the alarms and aggregating them at the centralized attack monitoring center allows for the use of locally generated alarms

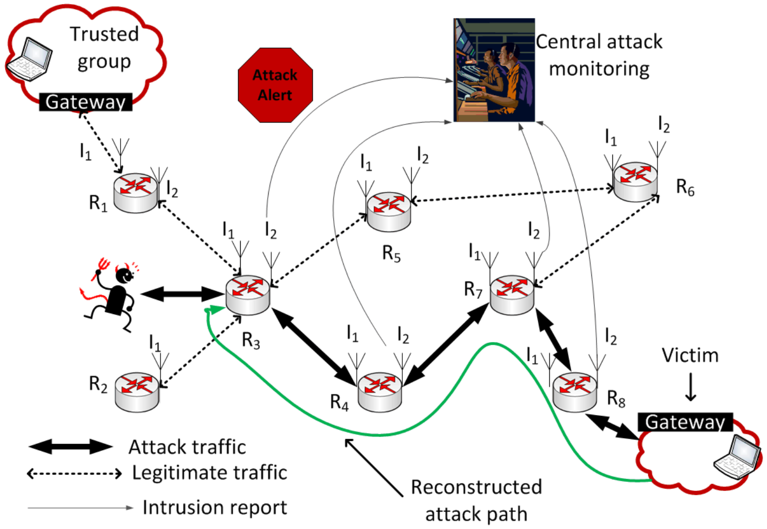


Fig. 7: We illustrate the mechanism to trace the source of the flooding attack in an anonymous network. Each router locally raises an alarm and informs the centralized attack monitoring center about the interface it is experiencing the attack. The attack path is constructed starting at the last node to report the attack and then the source is traced as R_3-R_4-R_7-R_8-$Victim$ for this example.

from possibly different phenomenons to construct a global view of the various routes under attack as shown in Fig. 7.

6 Conclusion and Future Work

With ubiquitous network access, users in trusted groups seek network anonymity for a variety of applications. Anonymous networks are still prone to network attackers who can disrupt communications. When attackers flood the network with packets, the routers of anonymous networks cannot distinguish flows of an attacker from the legitimate users. This makes traditional attack detection systems unusable for anonymous networks. In this work we designed a statistical distributed and hierarchical attack detection system for the anonymous networks using a readily observable metric. We proposed *packet resistance* as the metric which detected the presence of bandwidth attackers by observing the resistance to outgoing traffic at a router. The routers locally monitored for attack and raised alarms by indicating the interface they are experiencing a flooding attack's instance. These were collected by a centralized monitoring system to reconstruct the attack path and infer the attack's origin.

In our future work, we will design a larger experimental setup to demonstrate the effectiveness of the hierarchical decision aggregation using local statistical analysis at routers. We will then formalize our attack path reconstruction mechanism at the centralized attack monitoring center using alarms collected by aggregating the hierarchical decisions.

Acknowledgments

This research was supported through the Northrop Grumman Cybersecurity Research Consortium. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of Northrop Grumman or Carnegie Mellon University.

References

1. Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. Defense Technical Information Center (2000)
2. Dingledine, R., Mathewson, N., Syverson, P.: TOR: The second-generation onion router. Technical report, DTIC Document (2004)
3. Northcutt, S.: Network intrusion detection: An analyst's hand-book. EDPACS **27** (2000) 1–2
4. Ferguson, P.: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. (2000)
5. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers and Security **28** (2009) 18 – 28
6. Paxson, V.: Bro: a system for detecting network intruders in real-time. Computer Networks **31** (1999) 2435 – 2463
7. Wondracek, G., Holz, T., Kirda, E., Kruegel, C.: A practical attack to de-anonymize social network users. In: 2010 IEEE Symposium on Security and Privacy (SP),. (2010) 223–238
8. Murdoch, S., Danezis, G.: Low-cost traffic analysis of tor. In: Security and Privacy, 2005 IEEE Symposium on. (2005) 183–195
9. Back, A., Moller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In: Information Hiding. Lecture Notes in Computer Science. (2001) 245–257
10. Kyasanur, P., Vaidya, N.: Selfish mac layer misbehavior in wireless networks. Mobile Computing, IEEE Transactions on (2005) 502–516
11. Studer, A., Perrig, A.: The coremelt attack. In: Proceedings of the 14th European conference on Research in computer security. (2009) 37–52
12. Tor: Tor metrics portal: Graphs `https://metrics.torproject.org/graphs.html`.
13. Kang, M.S., Lee, S.B., Gligor, V.D.: The crossfire attack. In: 2013 IEEE Symposium on Security and Privacy (SP),. (2013) 127–141
14. Ahrenholz, J.: Comparison of CORE Network Emulation Platforms. In: IEEE MILCOM Conference,. (2010) 864–869
15. iperf: `http://iperf.sourceforge.net/`.