ISSN:0975-9646

Malicious Packet Losses Detection by using Compromised Router Detection Protocol

CHANDRA SEKHAR.G , SRIKANTH.D

Department of Computer Science and Engineering
Gudlavalleru Engineering College
Gudlavalleru — 521356

Abstract—**In this paper, we consider the problem of detecting whether a compromised router is maliciously manipulating its stream of packets. In particular, we are concerned with a simple yet effective attack in which a router selectively drops packets destined for some victim. Unfortunately, it is quite challenging to attribute a missing packet to a malicious action because normal network congestion can produce the same effect. Modern networks routinely drop packets when the load temporarily exceeds their buffering capacities. Previous detection protocols have tried to address this problem with a user-defined threshold: too many dropped packets imply malicious intent. However, this heuristic is fundamentally unsound; setting this threshold is, at best, an art and will certainly create unnecessary false positives or mask highly focused attacks. We have designed, developed, and implemented a compromised router detection protocol that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Once the ambiguity from congestion is removed, subsequent packet losses can be attributed to malicious actions. We have tested our protocol in Emu lab and have studied its effectiveness in differentiating attacks from legitimate network behavior**

Keywords— **Internet dependability, intrusion detection and tolerance, distributed systems, reliable networks, malicious routers.**

1.INTRODUCTION

THE Internet is not a safe place. Unsecured hosts can expect to be compromised within minutes of connecting to the Internet and even well-protected hosts may be crippled with denial-of-service (DoS) attacks. However, while such threats to host systems are widely understood, it is less well appreciated that the network infrastructure itself is subject to constant attack as well. Indeed, through combinations of social engineering and weak passwords, attackers have seized control over thousands of Internet routers [1], [2]. Even more troubling is Mike Lynn's controversial presentation at the 2005 Black Hat Briefings, which demonstrated how Cisco routers can be compromised via simple software vulnerabilities. Once a router has been compromised in such a fashion, an attacker may Interpose on the traffic stream and manipulate it maliciously to attack others—selectively dropping, modifying, or rerouting packets.

Several researchers have developed distributed protocols to detect such traffic manipulations, typically by validating that traffic transmitted by one router is received unmodified by another [3], [4]. However, all of these schemes—including our own—struggle in interpreting the absence of traffic. While a packet that has been modified in transit represents clear evidence of tampering, a missing packet is inherently ambiguous: it may have been explicitly blocked by a compromised router or it may have been dropped benignly due to network congestion.

In fact, modern routers routinely drop packets due to bursts in traffic that exceed their buffering capacities, and the widely used Transmission Control Protocol (TCP) is designed to cause such losses as part of its normal congestion control behavior. Thus, existing traffic validation systems must inevitably produce false positives for benign events and/or produce false negatives by failing to report real malicious packet dropping.

In this paper, we develop a compromised router detection protocol that dynamically infers the precise number of congestive packet losses that will occur. Once the congestion ambiguity is removed, subsequent packet losses can be safely attributed to malicious actions. We believe our protocol is the first to automatically predict congestion in a systematic manner and that it is necessary.

In the remainder of this paper, we briefly survey the related background material, evaluate options for inferring congestion, and then present the assumptions, specification, and a formal description of a protocol that achieves these goals. We have evaluated our protocol in a small experimental network and demonstrate that it is capable of accurately resolving extremely small and fine grained attacks.

2. BACKGROUND

There are inherently two threats posed by a compromised router. The attacker may subvert the network control plane (e.g., by manipulating the routing protocol into false route updates) or may subvert the network data plane and forward individual packets incorrectly. The first set of attacks has seen the widest interest and the most activity—largely due to their catastrophic potential. By violating the routing protocol itself, an attacker may cause large portions of the network to become inoperable. Thus, there have been a variety of efforts to impart authenticity and consistency guarantees on route update messages with varying levels of cost and protection [5], [6], [7], [8], [9], [10]. We do not consider this class of attacks in this paper.

Instead, we have focused on the less well appreciated threat of an attacker subverting the packet forwarding process on a compromised router. Such an attack presents a wide set of opportunities including (DoS), surveillance man-in-the-middle attacks, replay and insertion attacks, and so on. Moreover, most of these attacks can be trivially implemented via the existing command shell languages in commodity routers.

3. INFERRING CONGESTIVE LOSS

In building a traffic validation protocol, it is necessary to explicitly resolve the ambiguity around packet losses. Should the absence of a given packet be seen as malicious or benign? In practice, there are three approaches for addressing this issue:

- *Static Threshold*. Low rates of packet loss are assumed to be congestive, while rates above some predefined threshold are deemed malicious.
- *Traffic modeling*. Packet loss rates are predicted as a function of traffic parameters and losses beyond the prediction are deemed malicious.
- *Traffic measurement*. Individual packet losses are predicted as a function of measured traffic load and router buffer capacity. Deviations from these predictions are deemed malicious.

Most traffic validation protocols, including WATCHERS [3], Secure Trace route [12], and our own work described in [4], analyze aggregate traffic over some period of time in order to amortize monitoring overhead over many packets. For example, one validation protocol described in [4] maintains packet counters in each router to detect if traffic flow is not conserved from source to destination. When a packet arrives at router r and is forwarded to a destination that will traverse a path segment ending at router x, r increments an outbound counter associated with router x. Conversely, when a packet arrives at router r, via a path segment beginning with router x, it increments its inbound counter associated with router x. periodically, router x sends a copy of its outbound counters to the associated routers for validation. Then, a given router r can compare the number of packets that x claims to have sent to r with the number of packets it counts as being received from x, and it can detect the number of packet losses.

In order to avoid false positives, the threshold must be large enough to include the maximum number of possible congestive legitimate packet losses over a measurement interval. Thus, any compromised router can drop that many packets without being detected. Unfortunately, given the nature of the dominant TCP, even small numbers of losses can have significant impacts. Subtle attackers can selectively target the traffic flows of a single victim and within these flows only drop those packets that cause the most harm. For example, losing a TCP SYN packet used in connection establishment has a disproportionate impact on a host because the retransmission time-out must necessarily be very long (typically 3 seconds or more). Other seemingly minor attacks that cause TCP time-outs can have similar effects—a class of attacks well described in [17].

Because of some uncertainty in the system, we cannot predict exactly which individual packets will be dropped. So, our approach is still based on thresholds. Instead of being a threshold on rate, it is a threshold on a statistical measure: the amount of confidence that the drop was due to a malicious attack rather than from some normal router function. To make this distinction clearer, we refer to the statistical threshold as the target significance level.

4. SYSTEM MODEL

Our work proceeds from an informed, yet abstracted, model of how the network is constructed, the capabilities of the attacker, and the complexities of the traffic validation problem. In this section, we briefly describe the assumptions underlying our model. We use the same system model as in our earlier work [4].

4.1. NETWORK MODEL

We consider a network to consist of individual homogeneous routers interconnected via directional point-to point links. This model is an intentional simplification of real networks (e.g., it does not include broadcast channels or independently failing network interfaces) but is sufficiently general to encompass such details if necessary. Unlike our earlier work, we assume that the bandwidth, the delay of each link, and the queue limit for each interface are all known publicly.

Within a network, we presume that packets are forwarded in a hop-by-hop fashion, based on a local forwarding table. These forwarding tables are updated via a distributed link-state routing protocol such as OSPF or IS-IS. This is critical, as we depend on the routing protocol to provide each node with a global view of the current network topology. Finally, we assume the administrative ability to assign and distribute cryptography keys to sets of nearby routers. This overall model is consistent with the typical construction of large enterprise IP networks or the internal structure of single ISP backbone networks but is not well suited for networks that are composed of multiple administrative domains using BGP. At this level of abstraction, we can assume a synchronous network model.

4.2. THREAT MODEL

As explained in Section 1, this paper focuses solely on data plane attacks (control plane attacks can be addressed by other protocols with appropriate threat models such as [6], [7], [5], [8], [9], and [10]). Moreover, for simplicity, we examine only attacks that involve packet dropping.

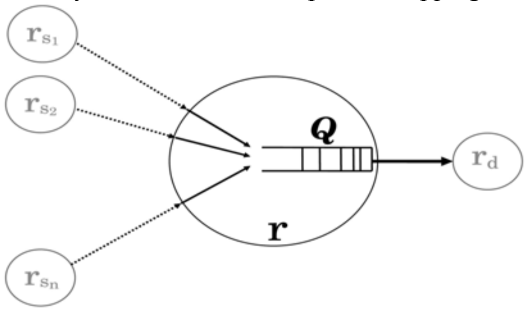

Fig.1. Validating the queue of an output interface.

However, our approach is easily extended to address other attacks—such as packet modification or reordering—similar to our previous work. Finally, as in [4], the protocol we develop validates traffic whose source and sink routers are uncompromised.

A router can be traffic faulty by maliciously dropping packets and protocol faulty by not following the rules of the detection protocol. We say that a compromised router r is traffic faulty with respect to a path segment _ during _ if _contains r and, during the period of time _, r maliciously drops or misroutes packets that flow through _. A router can drop packets without being faulty, as long as the packets are dropped because the corresponding output interface is congested. A compromised router r can also behave in an arbitrarily malicious way in terms of executing the protocol we present, in which case we indicate r as protocol faulty. A protocol faulty router can send control messages with arbitrarily faulty information, or it can simply not send some or all of them. A faulty router is one that is traffic faulty, protocol faulty, or both.

Attackers can compromise one or more routers in a network. However, for simplicity, we assume in this paper that adjacent routers cannot be faulty. Our work is easily extended to the case of k adjacent faulty routers.

χ PROTOCOL

Protocol χ tects traffic faulty routers by validating the queue of each output interface for each router. Given the buffer size and the rate at which traffic enters and exits a queue, the behavior of the queue is deterministic. If the actual behavior deviates from the predicted behavior, then a failure has occurred.

We present the failure detection protocol in terms of the solutions of the distinct sub problems: traffic validation, distributed detection, response.

5.1 TRAFFIC VALIDATION

The first problem we address is traffic validation: what information is collected about traffic and how it is used to determine that a router has been compromised.

Consider the queue Q in a router r associated with the output interface of link hr; rid (see Fig. 1). The neighbor routers rs1 rs2 , ... rsn feed data into Q .

In practice, the behavior of a queue cannot be predicted with complete accuracy. For example, the tulles in S and D may be collected over slightly different intervals, and so a packet may appear to be dropped when in fact it is not (this is discussed in Section 4.1). Additionally, a packet sent to a router may not enter the queue at the expected time because of short-term scheduling delays and internal processing delays.

5.1.1 SINGLE PACKET LOSS TEST

If a packet with fingerprint fp and size ps is dropped at time ts when the predicted queue length is qpred (ts), then we raise an alarm with a confidence value single, which is the probability of the packet being dropped maliciously.

The mean μ and standard deviation σ of X can be determined by monitoring during a learning period. We do not expect μ and σ to change much over time, because they are in turn determined by values that themselves do not change much over time. Hence, the learning period need not be done very often.

A malicious router is detected if the confidence value c_{single} is at least as large as a target significance level s^{evel}_{single}.

5.1.2 Combined Packet Losses Test

The second test is useful when more than one packet is dropped during a round and the first test does not detect a malicious router. It is based on the well-known Z-test4 [26].Let L be the set of n > 1 packets dropped during the last time interval. For the packets in L, let ps be the mean of the packet sizes, qpred be the mean of qpredðtsÞ (the predicted queue length), and qact be the mean of qactðtsÞ (the actual queue length) over the times the packets were dropped.

We test the following hypothesis: “The packets are lost due to malicious attack”: $\mu > q_{limit} - \overline{q_{pred}} - \overline{ps}$. The Z-test score is

$$z_1 = \frac{(q_{limit} - \overline{q_{pred}} - \overline{ps} - \mu)}{\sigma\sqrt{n}}.$$

For the standard normal distribution Z, the probability of *Prob(Z< z_1)* gives the confidence value $c_{combined}$ for the hypothesis. A malicious router is detected if $c_{combined}$ is at least as large as a target significance level $s^{level}_{combined}$.

One can question using a Z-test in this way because the set of dropped packets are not a simple random sample.But, this test is used when there are packets being dropped and the first test determined that they were consistent with congestion loss. Hence, the router is under load during the short period the measurement was taken and most of the points, both for dropped packets and for nondropped packets, should have a nearly full Q. In Section 7, we show that the Z-test does in fact detect a router that is malicious in a calculated manner.

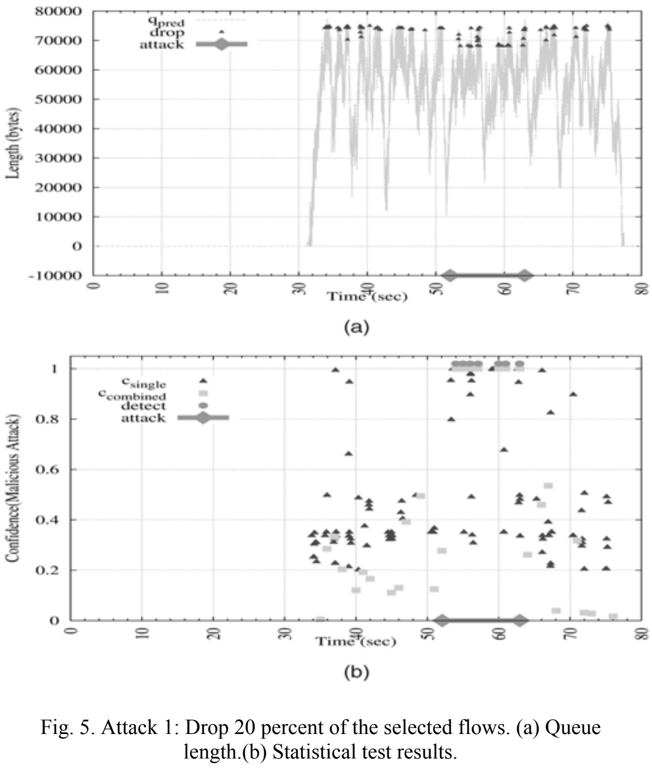

Fig. 5. Attack 1: Drop 20 percent of the selected flows. (a) Queue length.(b) Statistical test results.

6 ANALYSIS OF PROTOCOL χ

In this section, we consider the properties and overhead of protocol χ.

6.1 Accuracy and Completeness

In [4], we cast the problem of detecting compromised routers as a failure detector with accuracy and completeness properties. There are two steps in showing the accuracy and completeness of χ :

- Showing that TV is correct.
- Showing that χ is accurate and complete assuming that TV is correct.

6.2 Traffic Validation Correctness

Any failure of detecting malicious attack by TV results in a false negative, and any misdetection of legitimate behavior by TV results in a false positive.

Within the given system model of Section 4, the example TV predicate in Section 5.1 is correct. However, the system model is still simplistic. In a real router, packets may be legitimately dropped due to reasons other than congestion: for example, errors in hardware, software or memory, and transient link errors. Classifying these as arising from a router being compromised might be a problem, especially if they are infrequent enough that they would be best ignored rather than warranting repairs the router or link.

A larger concern is the simple way that a router is modeled in how it internally multiplexes packets. This model is used to compute time stamps. If the time stamps are incorrect, then TV could decide incorrectly. We hypothesize that a sufficiently accurate timing model of a router is attainable but have yet to show this to be the case.

A third concern is with clock synchronization. This version of TV requires that all the routers feeding a queue have synchronized clocks. This requirement is needed in order to ensure that the packets are interleaved correctly by the model of the router.

7 EXPERIENCES

We have implemented and experimented with protocol χ in the Emulab [35], [36] testbed. In our experiments, we used the simple topology shown in Fig. 3. The routers were Dell PowerEdge 2850 PC nodes with a single 3.0-GHz 64-bit Xeon processor and 2 Gbytes of RAM, and they were running Redhat-Linux-9.0 OS software. Each router except for r1 was connected to three LANs to which user machines were connected. The links between routers were configured with 3-Mbps bandwidth, 20-ms delay, and 75,000-byte capacity FIFO queue.

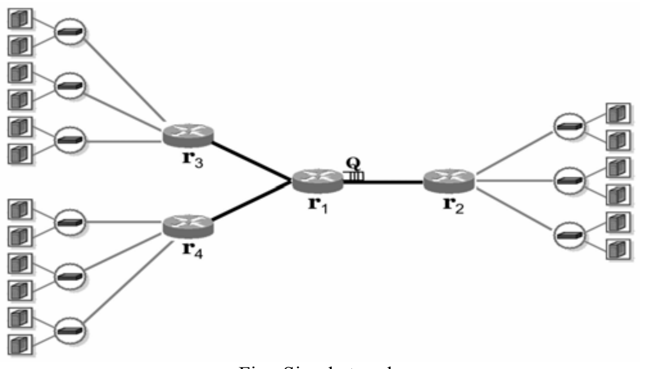

Fig.. Simple topology.

Each pair of routers shares secret keys; furthermore,integrity and authenticity against the message tampering is provided by message authentication codes.

The validation time interval *T* was set to 1 second, and the upper bound on the time to forward traffic information Δ was set to 300 ms. At the end of each second, the routers exchanged traffic information corresponding to the last validation interval and evaluated the TV predicate after 2Δ=600 ms. Each run in an experiment consisted of an execution of 80 seconds. During the first 30 seconds,we generated no traffic to allow the routing fabric to initialize. Then, we generated 45 seconds of traffic.

7.1 Experiment 1: Detecting Attacks

We then experimented with the ability of protocol χ to detect attacks. In these experiments, the router r1 is compromised to attack the traffic selectively in various ways, targeting two chosen ftp flows. The duration of the attack is indicated with a line bounded by diamonds in the figures, and a detection is indicated by a filled circle.

For the first attack, the router r1 was instructed to drop 20 percent of the selected flows for 10 seconds. Predicted queue length and the confidence values for each packet drop can be seen in Figs. 5a and 5b. As shown in the graph, during the attack, protocol χ detected the failure successfully.

In the second attack, router r1 was instructed to drop packets in the selected flows when the queue was at least 90 percent full. Protocol χ was able to detect the attack and raised alarms, as shown in Fig. 6.

8 NONDETERMINISTIC QUEUING

As described, our traffic validation technique assumes a deterministic queuing discipline on each router: FIFO with tail-drop. While this is a common model, in practice,real router implementations can be considerably more complex—involving switch arbitration, multiple layers of buffering, multicast scheduling, and so forth. Of these, the most significant for our purposes is the nondeterminism introduced by active queue management (AQM),such as random early detection (RED) [37], proportional integrator (PI) [38], and random exponential marking (REM) [39]. In this section, we describe how protocol χ can be extended to validate traffic in AQM environments.We focus particularly on RED, since this is the most widely known and widely used of such mechanisms.

RED was first proposed by Floyd and Jacobson in the early 1990s to provide better feedback for end-to-end congestion control mechanisms. Using RED, when a router's queue becomes full enough that congestion may be imminent, a packet is selected at random to signal this condition back to the sending host. This signal can take the form of a bit marked in the packet's header and then echoed back to the sender—Explicit Congestion Notification (ECN)[40], [41]—or can be indicated by dropping the packet.7 If ECN is used to signal congestion, then protocol χ , as presented in Section 5, works perfectly. If not, then RED will introduce nondeterministic packet losses that may be misinterpreted as malicious activity.

In the remainder of this section, we explain how RED's packet selection algorithm works, how it may be accommodated into our traffic validation framework, and how well we can detect even small attacks in a RED environment.

8.1 Random Early Detection

RED monitors the average queue size, q_{avg}, based on an exponential weighted moving average:

$$q_{avg} := (1-w)q_{avg} + w \cdot q_{act} \qquad (1)$$

where q_{act} is the actual queue size, and w is the weight for a low-pass filter.

RED uses three more parameters: q^{th}_{min}, minimum threshold;, q^{th}_{max} maximum threshold; and p_{max}, maximum probability.Using q_{avg}, RED dynamically computes a dropping probability in two steps for each packet it receives. First, it computes an interim probability, p_t:

$$p_t = \begin{cases} 0 & \text{if } q_{avg} < q^{th}_{min} \\ p_{max}\frac{q_{avg}-q^{th}_{min}}{q^{th}_{max}-q^{th}_{min}} & \text{if } q^{th}_{min} < q_{avg} < q^{th}_{max} \\ 1 & \text{if } q^{th}_{max} < q_{avg}. \end{cases}$$

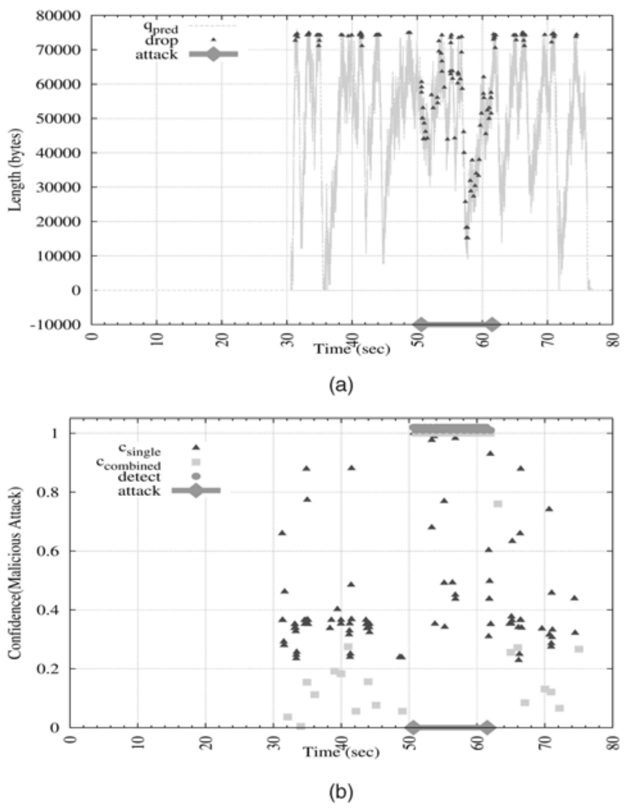

Fig.6.attack2: Drop the selected flows when the queue is 90 percent full. (a) Queue length. (b) Statistical test results.

Further, the RED algorithm tracks the number of packets,*cnt*, since the last dropped packet. The final dropping probability, *p*, is specified to increase slowly as *cnt* increases:

$$p = \frac{p_t}{1 - cnt \cdot p_t}. \quad (2)$$

Finally, instead of generating a new random number for every packet when $q^{th}_{min} < q_{avg} < q^{th}_{max}$, a suggested optimization is to only generate random numbers when a packet is dropped [37]. Thus, after each RED-induced packet drop, a new random sample, rn, is taken from a uniform random variable *R*= *Random*[0,1]. The first packet whose p value is larger than rn is then dropped, and a new random sample is taken.

8.2 Traffic Validation for RED

Much as in Section 5.1, our approach is to predict queue sizes based on summaries of their inputs from neighboring routers. Additionally, we track how the predicted queue size impacts the likelihood of a RED-induced drop and use this to drive two additional tests: one for the uniformity of the randomness in dropping packets and one for the distribution of packet drops among the flows.8 In effect,the first test is an evaluation of whether the distribution of packet losses can be explained by RED and tail-drop congestion alone, while the second evaluates if the particular pattern of losses (their assignment to individual flows) is consistent with expectation for traffic load.

9 ISSUES

9.1 Quality of Service

Real routers implement Quality of Service (QoS) providing preferential treatment to specified traffic via several different traffic-handling techniques, such as traffic shaping, traffic policing, packet filtering, and packet classification.

Given the configuration files, our work can be extended to handle these fairly complex real-life functions, even those involving nondeterminism, if the expected behavior of the function can be modeled.

9.2 Adjacent Faulty Routers

We assume that there exists no adjacent faulty routers in our threat model for simplicity. This assumption eliminates consorting faulty routers that collude together to produce fraudulent traffic information in order to hide their faulty behavior. However, it can be relaxed to the case of $k > 1$ adjacent faulty routers by monitoring every output interface of the neighbors k hops away and disseminating the traffic information to all neighbors within a diameter of k hops. This is the same approach that we used in [4], and it increases the overhead of detection.

10 CONCLUSION

To the best of our knowledge, this paper is the first serious attempt to distinguish between a router dropping packets maliciously and a router dropping packets due to congestion.Previous work has approached this issue using a static user-defined threshold, which is fundamentally limiting.Using the same framework as our earlier work (which is based on a static user-defined threshold) [4], we developed a compromised router detection protocol χ that dynamically infers, based on measured traffic rates and buffer sizes, the number of congestive packet losses that will occur. Subsequent packet losses can be attributed to malicious actions. Because of nondeterminism introduced by imperfectly synchronized clocks and scheduling delays,protocol χ uses user-defined significance levels, but these levels are independent of the properties of the traffic.Hence, protocol χ does not suffer from the limitations of static thresholds.

We evaluated the effectiveness of protocol χ through an implementation and deployment in a small network. We show that even fine-grained attacks, such as stopping a host from opening a connection by discarding the SYN packet, can be detected.

ACKNOWLEDGMENTS

The first author would like to thank Mustafa Arisoylu for the insightful discussions about the dynamics of RED algorithm. This material is based upon Alper T. M_zrak's dissertation research at University of California, San Diego.

REFERENCES

[1] X. Ao, Report on DIMACS Workshop on Large-Scale Internet Attacks,http://dimacs.rutgers.edu/Workshops/Attacks/internet-attack-9-03.pdf, Sept. 2003.

[2] R. Thomas, ISP Security BOF, NANOG 28, http://www.nanog.org/mtg-0306/pdf/thomas.pdf, June 2003.

[3] K.A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R.A. Olsson, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach," Proc. IEEE Symp. Security and Privacy (S&P '98), pp. 115-124, May 1998.

[4] A.T. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage, "Detecting and Isolating Malicious Routers," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 3, pp. 230-244, July-Sept. 2006.

[5] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and Whisper: Security Mechanisms for BGP," Proc. First Symp.Networked Systems Design and Implementation (NSDI '04), Mar. 2004.

[6] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (Secure-BGP)," IEEE J. Selected Areas in Comm.,vol. 18, no. 4, pp. 582-592, Apr. 2000.

[7] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Proc. ACM MobiCom '02, Sept. 2002.

[8] B.R. Smith and J. Garcia-Luna-Aceves, "Securing the Border Gateway Routing Protocol," Proc. IEEE Global Internet, Nov. 1996.

[9] S. Cheung, "An Efficient Message Authentication Scheme for Link State Routing," Proc. 13th Ann. Computer Security Applications Conf.(ACSAC '97), pp. 90-98, 1997.

[10] M.T. Goodrich, Efficient and Secure Network Routing Algorithms,provisional patent filing, Jan. 2001.

[11] R. Perlman, "Network Layer Protocols with Byzantine Robustness," PhD dissertation, MIT LCS TR-429, Oct. 1988.

[12] V.N. Padmanabhan and D. Simon, "Secure Traceroute to Detect Faulty or Malicious Routing," SIGCOMM Computer Comm. Rev.,vol. 33, no. 1, pp. 77-82, 2003.

[13] I. Avramopoulos and J. Rexford, "Stealth Probing: Efficient Data-Plane Security for IP Routing," Proc. USENIX Ann. Technical Conf.(USENIX '06), June 2006.

[14] S. Cheung and K.N. Levitt, "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection,"Proc. Workshop on New Security Paradigms (NSPW '97), pp. 94-106,1997.

[15] J.R. Hughes, T. Aura, and M. Bishop, "Using Conservation of Flow as a Security Mechanism in Network Protocols," Proc. IEEE Symp. Security and Privacy (S&P '00), pp. 131-132, 2000.

[16] A. Mizrak, Y. Cheng, K. Marzullo, and S. Savage, "Fatih: Detecting and Isolating Malicious Routers," Proc. Int'l Conf. Dependable Systems and Networks (DSN '05), pp. 538-547, 2005.