RESEARCH ARTICLE OPEN ACCESS

DETECTING MALICIOUS PACKET LOSSES

I. Dhanaseeli

Assistant Professor, Department of Computer Science, Immaculate College for Women, Viriyur.

Abstract:

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. 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 Emulab and have studied its effectiveness in differentiating attacks from legitimate network behavior.

Keywords:

I. 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. 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.

II.EXISTING SYSTEM:

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.

The earliest work on fault-tolerant forwarding is due to Pearlman who developed a robust routing system based on source routing, digitally signed route-setup packets, and reserved buffers.

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.

III.PROPOSED SYSTEM:

In contrast, protocol X can detect such malicious behaviors because it measures the router's queues, which are determined by the dynamics of the network transport protocol. Protocol _ can report false positives and false negatives, but the probability of such detections can be controlled with a significance level for the statistical tests upon

which is built. A static threshold cannot be used in the same way.

To summarize, these protocols are designed to detect anomalies between pairs of correct nodes, and thus for simplicity, it is assumed that a terminal router is not faulty with respect to traffic originating from or being consumed by that router.

IV.MODULES

1. **Create Network Environment**
2. **Packet Collection Operation.**
3. **Packet forwarding using Static Threshold.**
4. **Selection of congested area**
5. **Packet forwarding using Compromised router Detection Protocol.**

Module 1

In first module, at first we create an environment. The environment setup can be in rectangular area. The nodes in the environment can be aligned in Random Access method. It means each node consists of four neighbor nodes. It can be fixed through mesh topology.

Module 2

In this module packet collection process was done between the aggregated nodes and the member nodes. Using the coverage distance and timing events.

Module 3

In this module described about forwarding the packets using buffer size. It defines the node number, size of the packet and the packet loss using the static threshold method.

V. CONCLUSIONS

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, we developed a compromised router detection protocol x 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 no determinism introduced by imperfectly synchronized clocks and scheduling delays, protocol x uses user-defined significance levels, but these levels are independent of the properties of the traffic.

VI. REFERENCES

[1] Chieh-Jen Cheng, Chao-Ching Wang, Wei-Chun Ku, Tien-Fu Chen , and Jinn-Shyan Wang, “Scalable High-Performance Virus Detection Processor Against a Large Pattern Set for Embedded Network Security” Commun. vol. 51, pp. 62–70,2011.

[2] O. Villa, D. P. Scarpazza, and F. Petrini, “Accelerating real-time string searching with multicore processors,” Computer, vol. 41, pp. 42–50,2008.

[3] D. P. Scarpazza, O. Villa, and F. Petrini, “High-speed string searching against large dictionaries on the Cell/B.E. processor,” in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2008, pp. 1–8.

[4] D. P. Scarpazza, O. Villa, and F. Petrini, “Peak-performance DFA based string matching on the Cell processor,” in Proc. IEEE Int. Symp. Parallel Distrib. Process., 2007, pp. 1–8.

[5] L. Tan and T. Sherwood, “A high throughput string matching architecture for intrusion detection and prevention,”in Proc. 32nd Annu. Int. Symp. Comput. Arch., 2005, pp. 112–122.

[6] S. Dharmapurikar, P. Krishnamurthy, and T. S. Sproull, “Deep packet inspection using parallel bloom filters,” IEEE Micro, vol. 24, no. 1, pp.52–61, Jan. 2004.

[7] R.-T. Liu, N.-F. Huang, C.-N. Kao, and C.-H. Chen, “A fast string matching algorithm for network processor-based intrusion detection system,” ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[8] F. Yu, R. H. Katz, and T. V. Lakshman, “Gigabit rate packet pattern matching using TCAM,” in Proc. 12th IEEE Int. Conf. Netw. Protocols, 2004, pp. 174–178.intrusion

detection system," ACMTrans. Embed. Comput. Syst., vol. 3, pp. 614–633, 2004.

[9] R. S. Boyer and J. S. Moore, "A fast string searching algorithm,"Commun. ACM, vol. 20, pp. 762–772, 1977.

[10] V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliographic search," Commun. ACM, vol. 18, pp. 333–340, 1975