Network Management Framework for Network Forensic Analysis

Ankita Bhondele[1], Shatrunjay Rawat[2], and Shesha Shila Bharadwaj Renukuntla[2]

[1] Bansal Institute of Science and Technology, Bhopal, India
ankitabhondele@gmail.com
[2] International Institute of Information Technology, Hyderabad, India
shatrunjay.rawat@iiit.ac.in,
sheshashila.bharadwaj@reserach.iiit.ac.in

Abstract. Tracing malicious packets back to their respective sources is important to defend the internet against attacks. Content based trace-back techniques have been proposed to solve the problem of source identification. It is not feasible to effectively store and query all the data stored in the devices for extended periods of time due to resource limitations in the network devices.

In this paper, we propose a management framework for network packet trace-back with optimum utilization of device storage capacity. We aim to remotely manage the devices and also to store large forensic data so that we can identify the source of even older attacks.

Keywords: Network Management Framework, Bloom filters, Network forensics.

1 Introduction

Computer network has experienced a rapid growth over the past years and with this growth have also come several security issues. To secure network from different types of attacks, attempts have been made at hardware and software levels. New security applications, Firewalls, Antivirus Softwares, Intrusion Detection Systems and many more are used to protect network and host from damage. To stop attacks and their side-effects, it is important to know the actual source of attack so that we may block the malicious system creating problem.

Previously, source machine was identified based on IP address in the packet header which can be altered or spoofed. This is not an effective way to identify the actual source of packet. Second way of source identification is based on packet payload. To perform trace-back in a network based on payload, payload attribution has to be performed by each network device through which packet passes. Payload attribution [1], [6] is an important element in network forensics, which makes the identification of source of packet possible, based on part of packet payload called "*excerpt*". Practically, it is not possible to store the entire payload at every routing device due to storage and privacy concerns. To deal with the above problem, it is required that data

S.C. Satapathy et al. (eds.), *Emerging ICT for Bridging the Future – Volume 2,*
Advances in Intelligent Systems and Computing 338, DOI: 10.1007/978-3-319-13731-5_43

be stored in compressed form. Burton Howard Bloom, in 1970, proposed a bloom filter [4] which is a space efficient data structure that works on probabilistic algorithm. Bloom Filter has its application in various fields including network forensics [5].Various hash based techniques and their variations have been proposed [2] for storing packet content which are Source Path Isolation Engine (SPIE) [7], Block Bloom Filters (BBF) and Hierarchical Bloom Filters (HBF) to allow analysis of network events for investigation purposes.

Implementation of same payload attribution techniques on all routers is practically not possible as devices have different storage and processing capacity. New technique was proposed [3] which provides flexible environment for the implementation of these attribution techniques based on parameters like block size and false positive probability (FPP). This heterogeneous implementation saves storage space as well as reduces the processing burden on devices. With increase in network traffic, bloom filters required for storing forensic data at each device will also increase. In today's networks, it is difficult to store and query all the packet data for extended periods of time. To offer more protection to the network from damage, we will require forensic data to be stored for longer period which results into larger storage space requirement for devices.

In this paper, we propose a framework that solves the problem of storage requirement at devices during forensic analysis and allows remote management of devices. The proposed management framework uses Simple Network Management Protocol (SNMP) to communicate between Network Management System (Manager) and network device (Agent). During trace-back, each device in this framework will implement payload attribution method with varying block size and False Positive Probability as per requirement. We have proposed methods under Network Management Framework that allow us to store large amount of data for longer period so that we can investigate even an older attack and find out the source for the same. Due to limited storage, it is difficult for devices to store large amount of information. The Network Management System (NMS) based framework overcomes the problem of storage space requirement at each network device by storing all the information at NMS end.

2 Proposed Network Management Framework

For the source identification, content based trace-back techniques were introduced over traditional trace-back techniques. Content based techniques overcome the problem of storage requirement for devices by storing data in compact form using "*bloom filters*". In a Network, multiple machines may face attacks. In a large network, it is not possible to trace-back using single process as it will consume time. We need an approach that will identify source of attack using multi-process attributionsystem so that network can be protected from attacks.We propose a framework that provides a set of management services which help during forensic analysis.

The proposed Network Management framework consists of two main components: Network Management System (manager) and Device (Agent). SNMP is used to remotely manage a large network. This management protocol is used between the NMS and the device to make management related communication possible. The Request-Response nature of the protocol plays a big role in management tasks. The framework will monitor network activity and manage network resources as well.

The proposed work will provide network visibility to the administrator, so that he can perform trace-back for any victim based on malicious content identified. We have also introduced methods for efficient utilization of available storage space at devices to make payload attribution system more feasible. We have aimed towards minimization of storage space required during trace-back process. The methods under proposed framework are classified based on the way forensic data is stored.

2.1 Methods for Storing Forensic Data in a Network

To investigate instance of attack, information stored in devices plays a big role. Thus, the process of storing forensic data needs more attention. In order to make payload attribution system more scale-able, we have given methods under the framework for storing forensic data during analysis. In addition to the reduction of storage space, proposed methods also reduce processing burden on devices. There are two methods in the NMS based management framework for storing and processing forensic data which are named as *Distributed Management* and *Centralized Management methods.*

Distributed Management Method. In a distributed management, as the host receives any malicious packet from network, it informs the central authority about an occurred attack as shown in Fig.1. Upon receiving an “excerpt” (part) of malicious packet from victim host, NMS takes the responsibility of taking required action. In trace-back mechanism, each device stores forensic data in a special data structure called “Bloom Filter” using parameterized approach. Depending upon the device capability, we can implement BBF and HBF.

NMS sends query to devices to perform traceback in network for identifying the attacker, using the excerpt of packet. Device receives the excerpt query from NMS and checks whether excerpt is present in the available Bloom filters or not. To determine the path taken by the malicious packet, the query has to be matched with the bloom filter content at every device during traceback process. If the query matches stored bloom filter content, it means that the malicious packet has passed through that device. Each device in a network is configured to send query response back to NMS. NMS records all these query responses for future use. After performing membership test, query is sent to another device. Based on received response from each device, NMS constructs the path traversed by malicious packet. Once the path and source of attack is identified by NMS, it can take appropriate action against the attacker for future security. In such a way, each device is checked during traceback by NMS in a distributed management framework for identifying the source of attack.

In distributed management method, payload attribution works in way similar to other techniques as discussed in section II, but under the supervision of Management system. This method does not work for the minimization of storage space requirement. If we have all the bloom filters stored at a single device, failure of the device may result in loss of all forensic data. To avoid such risk, it is beneficial to have distributed forensic data.

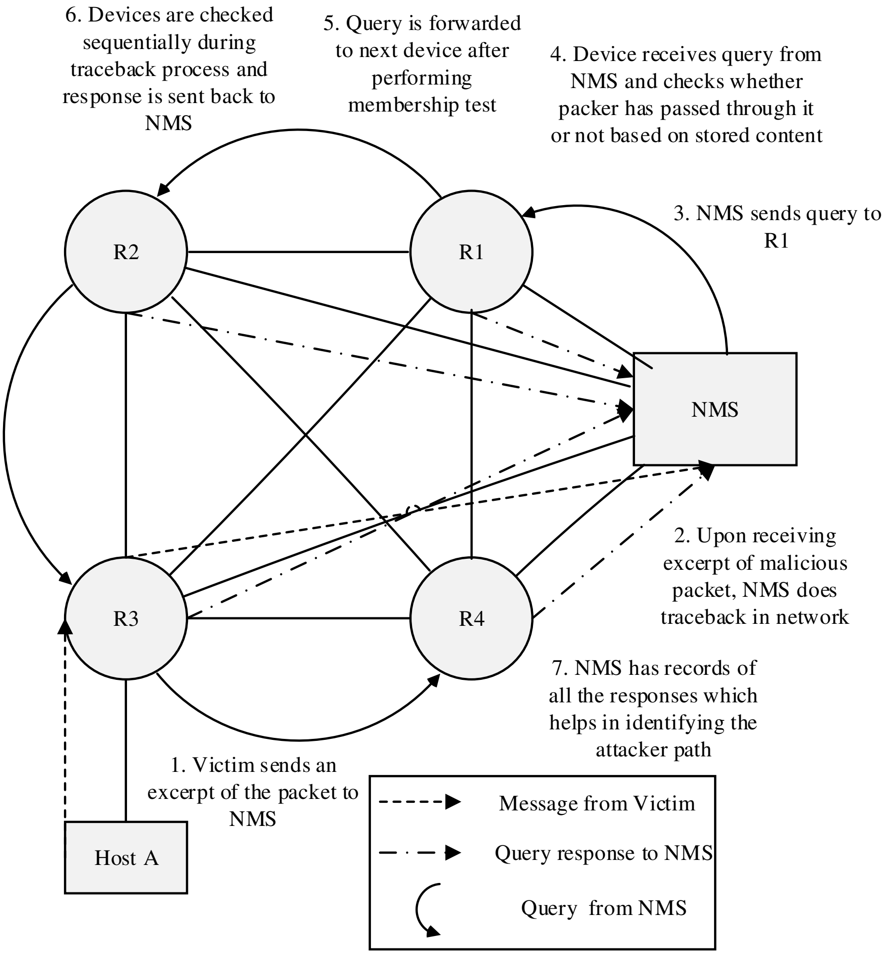
6. Devices are checked sequentially during traceback process and response is sent back to NMS
5. Query is forwarded to next device after performing membership test
4. Device receives query from NMS and checks whether packer has passed through it or not based on stored content
3. NMS sends query to R1
R2
R1
NMS
R3
R4
2. Upon receiving excerpt of malicious packet, NMS does traceback in network
7. NMS has records of all the responses which helps in identifying the attacker path
1. Victim sends an excerpt of the packet to NMS
Host A
Message from Victim
Query response to NMS
Query from NMS

Fig. 1. Distributed Management Method

Centralized Management Method. Each device in second method, as shown in Fig. 2, is configured to store forensic data. For attack analysis, all the forensic data stored in devices is sent to a central storage to avoid distributed processing. Device sends stored forensic data periodically to NMS, thus we have all the bloom filters at one place. In a centralized management method, upon receiving an excerpt of malicious packet, NMS does not send query to network devices to check whether malicious packet has passed through it or not. Instead of querying network devices for forensic data, NMS itself processes all the queries and performs membership test on the data collected from network devices. All the distributed forensic data is recorded by NMS at one place and this recorded information helps NMS in reducing the packet and processing load of the network due to querying process.

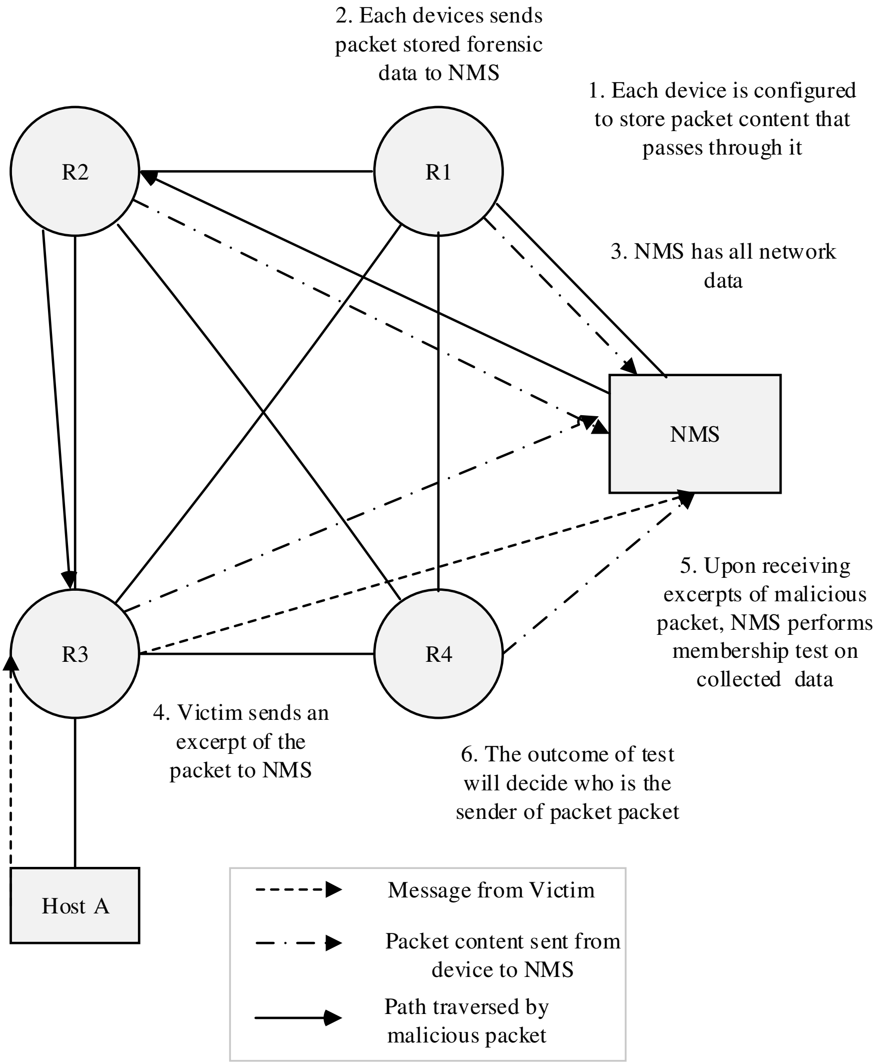
2. Each devices sends packet stored forensic data to NMS
1. Each device is configured to store packet content that passes through it
R2
R1
3. NMS has all network data
NMS
5. Upon receiving excerpts of malicious packet, NMS performs membership test on collected data
R3
R4
4. Victim sends an excerpt of the packet to NMS
6. The outcome of test will decide who is the sender of packet packet
Host A
Message from Victim
Packet content sent from device to NMS
Path traversed by malicious packet

Fig. 2. Centralized Management Method

NMS performs the membership test for identifying the source of packet and test results will decide the path taken by the attack packet. For storing forensic data centrally, it is required to know how to extract stored forensic data from devices. For this purpose we have introduced methods so that the manager can easily extract information from agent (device). These methods are named as Pull Process and Push Process.

Pull Process. Once the bloom filter saturates, it needs to be moved to NMS storage. For the purpose of storing forensic data centrally, manager periodically polls the devices. This polling process works as per predefined intervals. Thus, deciding the interval of querying becomes an important part of this process. Network is dynamic in nature and consists of devices ranging from high speed backbone router to simple edge router. In backbone router, the information update takes place more frequently as compared to an edge router which demands a shorter query interval. It is not practically possible to define a separate query interval for each device. To avoid loss of information, the interval should be chosen in such a way that it is neither too longnor too short.

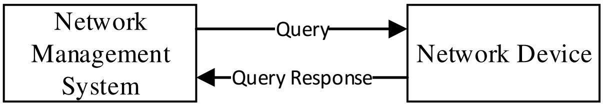

Fig. 3. Pull Process

Push Process. In push process, saturated bloom filters are pushed by devices to manager. This push process uses special type of message called "**Trap**". This type of message is used by SNMP for event notification. *We can use Trap to notify an event such*as saturation of bloom filter, memory overflow, etc. so that manger can take required action immediately to avoid data loss. *This process also saves time and effort*. We aim to introduce storage efficient payload attribution system using NMS based push process. For pushing bloom filters to manger, we have given two methods classified based on request-response (exchange between NMS and device) sent. First is "**Trap"** Only method and second method is Data in "**Trap"**.

In the first type of push process, agent (device) will send Trap notification to NMS informing about the saturation of bloom filters or occurrence of memory overflow. Upon receiving notification from device, NMS sends query back to devices to get the instances of an occurred event. After getting value from device, NMS can take appropriate action in order to solve the problem so that device can start again for storing forensic data for attack identification.

In the second type, notification about saturation of bloom filters is sent by devices to NMS along with its value. It saves bandwidth as single message is required for sending notification and data both. For reducing storage space requirement at devices, the process of sending bloom filters along with the "Trap" is used under push process. This process also saves time and processing for data extraction. Thispossible to send bloom filter with in the "Trap" by placing its value in the variable-binding field.

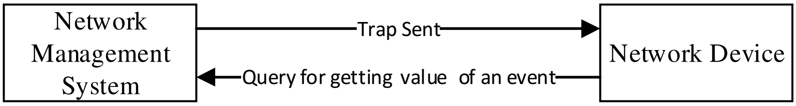
Network Management System
Trap Sent
Query for getting value of an event
Network Device

Fig. 4. TrapOnly

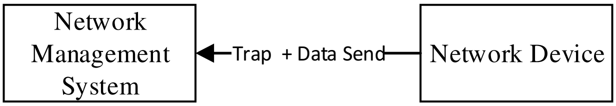

Fig. 5. Data in Trap

Benefits of Push Process over Pull Process. In a pull process under management framework, the manager polls devices for forensic data. Every time the manager needs an update, all the devices respond whether they have any updates or not. This results into burst of traffic, as manager sends a query for updated bloom filters to all network devices. By reducing the frequency of the polling process, we can reduce network traffic which in turn reduces network visibility. This reduced network visibility increases the risk of high forensic data loss, as most of the notifications go unanswered. Due to the loss of forensic data, we may fail to identify the source of attack. In a large network, it becomes impractical for manager to poll a large number of devices, which results in increased traffic and effort by the manager. The push process is a solution to the problem because it does not require a request from the manager. It sends message only when the device changes its state when bloom filters saturates. This process results in a substantial reduction in network traffic and allows data to be sent within a message. This also reduces storage space requirement at devices.

Comparison between Centralized and Distributed Management Methods

We have compared centralized and distributed methods, to analyze which method is more appropriate for effective payload attribution, based on few parameters listed below:

Storage and Processing burden: In distributed method, forensic data stored in the devices will remain in the device itself throughout the analysis process. For checking whether attack packet has passed through the device or not, device has to perform membership test which adds an extra processing burden on devices. Increase in the number of forensic queries will also result in increased processing load on devices during trace-back process. On other hand, in centralized management, all the forensic data is brought to NMS and processed there. This saves excess processing.

In distributed method, each device has to store large amount of forensic data in bloom filters for investigation purposes. Each device has limited memory and it cannot store forensic content for longer periods. The purpose of introducing centralized management method is to decrease the storage requirement by efficiently using available resources at device's end. NMS used in this method supports various functionalities and it can store huge amount of forensic data for desired period of time.

Network traffic: In distributed management,NMS has to send forensic queries to devices, as forensic data is in distributed form which results in increased network traffic.

*Time consumption:*It is difficult to implement distributed method in a larger network, as trace-back process may take lot of time as we have to check all the connected devices in the network topology. It is suitable to use distributed management in smaller networks as querying each device will not take much time.

Implementing distributed management in a network for processing and storing forensic data is not a practical solution as network consists of devices that vary in their processing and storage capacity. Thus, all the devices will not be able to store same amount of forensic data and will not support same processing speed. Processing speed and storage capacity are two important parameters and cannot be compromised. The centralized management method overcomes these problems of storing and processing data, and provides network visibility to the network administrator.

3 Conclusion

In this paper, we have shown how large storage space requirement during forensic analysis at devices can be reduced with the help of proposed framework. This framework consists of management system that manages all the network components and activities. SNMP is used for the information exchange and provides additional features that make the framework more effective for the identification of source of attack. We have given two methods under Framework namely distributed and centralized method. Using centralized method, we are able to store more data for forensic analysis. In this way, the proposed framework minimizes the processing and storage burden on network devices which was there in the previous methods. In future, this work can be extended by implementing the proposed methods to test its feasibility.

References

1. Shanmugasundaram, K., Bronnimann, H., Memon, N.: Payload attribution via hierarchical Bloom filters. In: CCS 2004: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 31–41. ACM, New York (2004)
2. Ponec, M., Giura, P., Bronnimann, H., Wein, J.: Highly efficient technique for network forensics. In: The ACM Computer and Communication Security Conference (2007)
3. Shujath, M.S., Rawat, S.: Heterogeneous Configuration of Bloom Filter for Network Forensic Analysis. In: IEEE-CYBER (2012)
4. Bloom, B.: Space/Time Trade-Offs in Hash Coding with Allowable Errors. Comm. ACM 13(7), 422–426 (1970)
5. Broder, A.Z., Mitzenmacher, M.: Network applications of bloom filters: A survey. In: Fortieth Annual Allerton Conference on Communication, Control, and Computing, Coordinated Science Laboratory and the Department of Electrical and Computer Engineering of the University of Illinois at Urbana-Champaign (2002)
6. Shanmugasundaram, K., Memon, N., Savant, A., Bronnimann, H.: Fornet: A distributed forensics network. In: Workshop on Mathematical Methods, Models, and Architectures for Computer Networks Workshop, MMM-ACNS (2003)
7. Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T., Strayer, W.T.: Hash-based IP traceback. In: ACM SIGCOMM, SanDiego, California, USA (August 2001)