International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 4 Issue: 1 26-Nov-2014,ISSN_NO: 2321-3337

A survey on alert based link error reduction and priority based download approach in MANET

[1]R. AYYAPPAN, [2]R. JAYASHREE, [3]M. RAMESH KUMAR.

[1, 2] M.E Student, [3]Assistant Professor.

[1, 2, 3] Vel Tech Multi Tech Dr.Rangarajan Dr.Sakunthala Engineering College,

Chennai, Tamil Nadu, India.

ABSTRACT– *MANET is a self-configuring network in which mobile nodes are connected by wireless link. When a user is trying to download a file from the network, he may or may not able to download the file. Due to the link errors and malicious packet dropping are the two sources for packet losses in the wireless adhoc network. To ensure the truthful calculation of these errors, they have developed a homomorphic linear authenticator (HLA) based public auditing architecture to report the packet loss information by the nodes. Since the architecture is complexity and high cost to detect the errors and the system can only monitor the current state problem? In order to improve the detection accuracy, we propose the tool called the windows packet capture (winpcap). The purpose of the tool is to give alert to the user about the link errors and the packet drop. And, we will also create a log at the backend, so that the user who request the file, will be given an priority, so based on the priority the system will download the file to the user. The purpose of the log is to make the network a collision and traffic proof.*

Key terms: **packet dropping, secure routing, link error, auditing, winpcap tool.**

1, INTRODUCTION

In a multi-hop wireless network [1], nodes cooperate in relaying/ routing traffic. An adversary can exploit this cooperative nature to launch attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery [3] process. Once being included in a route, the adversary starts dropping packets [2],[3],[5]. In the most severe form, the malicious node simply stops forwarding every packet received from upstream nodes, completely disrupting the path between the source and the destination. Even though persistent packet dropping can effectively degrade the performance of the network [5], from the attacker's standpoint such an "always-on" attack has its disadvantages. First, the continuous presence of extremely high packet loss rate [7] at the malicious nodes makes this type of attack easy to be detected. Second, once being detected, these attacks are easy to mitigate.

If the malicious nodes [8] are also identified, their threats can be completely eliminated by simply deleting these nodes from the network's routing [4] table. A malicious node[1],[6][8] that is part of the route can exploit its knowledge of the network protocol and the communication context to launch an *insider attack*–an attack[7] that is intermittent, but can achieve the same performance degradation effect as a persistent attack at a much lower risk of

being detected. Specifically, the malicious node may evaluate the importance of various packets [4], and then drop the small amount that are deemed highly critical to the operation of the network.

In this paper, we are interested in combating such an insider attack. In particular, we are interested in the problem of detecting the occurrence of selective packet drops and identifying the malicious node(s) [1] responsible for these drops. Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped [5], but also identify whether the drop is intentional or unintentional. Specifically, due to the open nature of wireless medium, a packet drop in the network[6] could be caused by harsh channel conditions (e.g., fading, noise, and interference, a.k.a., link errors), or by the insider attacker. In an open wireless environment, link errors are quite significant, and may not be significantly smaller than the packet dropping [2] rate of the insider attacker. So, the insider attacker [8] can camouflage under the background of harsh channel conditions. In this case, just by observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss [3].

1.1 Objective:

The main aim of the project is to rectify the link errors [1], [3] and make the download easy with a single request.

1.2 Scope:

The scope of this paper is, by using the tool [12], [13], it will give alert to the user about the errors and the packet drop which are reported by the nodes. We can monitor the entire LAN for the packet loss.

2. SYSTEM ANALYSIS

System Analysis is a combined process dissecting the system responsibilities that are based on the problem domain characteristics and user requirements.

2.1 Existing System:

Link errors and malicious packet dropping [1], [4] are two sources for packet losses in wireless adhoc network. While observing a sequence of packet losses [3] in the network, it is determining whether the losses are caused by link errors only or the combination of both the link errors and malicious packet drops. To ensure the truthful calculation of these errors, they have developed a homomorphic linear authenticator (HLA)[1] based public auditing architecture that allows the detector to verify the truthfulness of the packet loss information reported by nodes[2]. We found that it is critical to acquire truthful packet loss information [5],

[6] at each individual nodes[7],[8], and it requires high computational cost and storage overheads. Through the ns2 simulator they have showed the packet loss information.

Disadvantages:

- User needs to give multiple requests to the server.
- Using the architecture, it requires high computational cost and storage overhead.
- Persistent packet dropping can effectively degrade the performance of the network.

2.2 Proposed System:

To improve the detection accuracy, we propose to exploit the correlations between the lost packets. Furthermore to ensure the truthful calculation of packet loss[13], we have proposed a tool named winpcap[12], that will give an alert to the user about the errors and the packet drop[13] which are reported by the nodes. And, we have maintained the log at the backend, so the user who request the file to download will be given the priority. Using the clustering algorithm and batch auditing technique we can collect the similar data together and send to the correct user[12]. In the proposed system, we have maintained the three attributes such as the request, proceed and download.

Advantages:

- It stops the multiple request from the user to download a file.
- Using the tool, it will give alert to the user about the errors and the packet drop which are reported by the nodes.
- We can monitor the entire LAN for the packet loss.
- By creating the log, timeliness – multiple data delivery speed options.
- By creating the log, reliability – multipath forwarding.

3. SYSTEM DESIGN

System Design involves identification of classes their relationship as well as their collaboration. In object or, classes are divided into entity classes and control classes. The Computer Aided Software Engineering (CASE) tools that are available commercially do not provide any assistance in this transition. CASE tools take advantage of Meta modeling that is helpful only after the construction of the class diagram. In the FUSION method some object-oriented approach likes Object Modeling Technique (OMT), Classes, and Responsibilities. Collaborators (CRC), etc, are used. Object used the term" agents" to represent some of the hardware and software system. In Fusion method, there is no requirement phase, where a user will supply the initial requirement document. Any software paper is worked out by both the analyst and th e designer. The analyst creates the user case diagram. The designer creates the class diagram. But the designer can do this only after the analyst creates the use case diagram. Once the design is over, it is essential to decide which software is suitable for the application.

3.1 ARCHITECTURAL DIAGRAM:

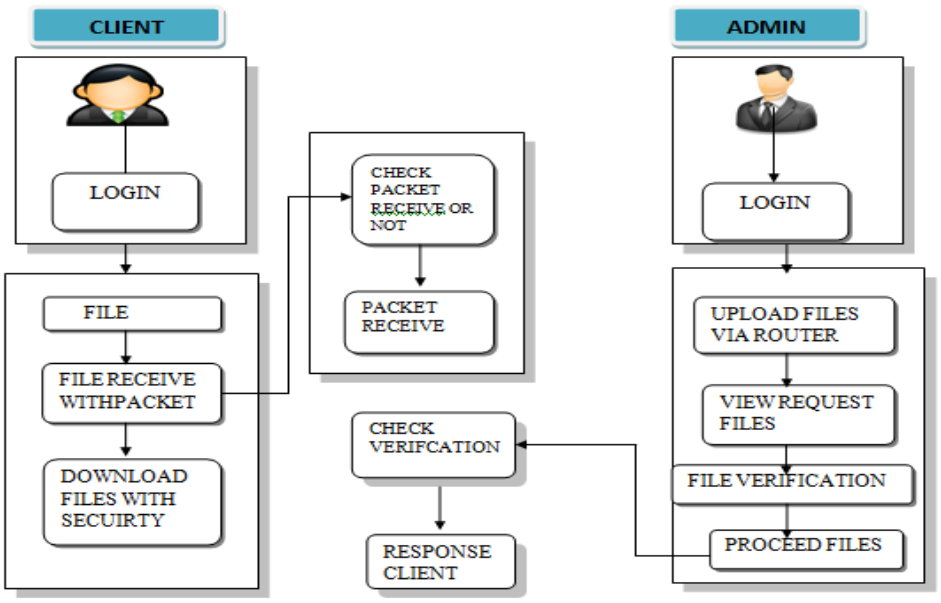

Fig:1 Priority based download.

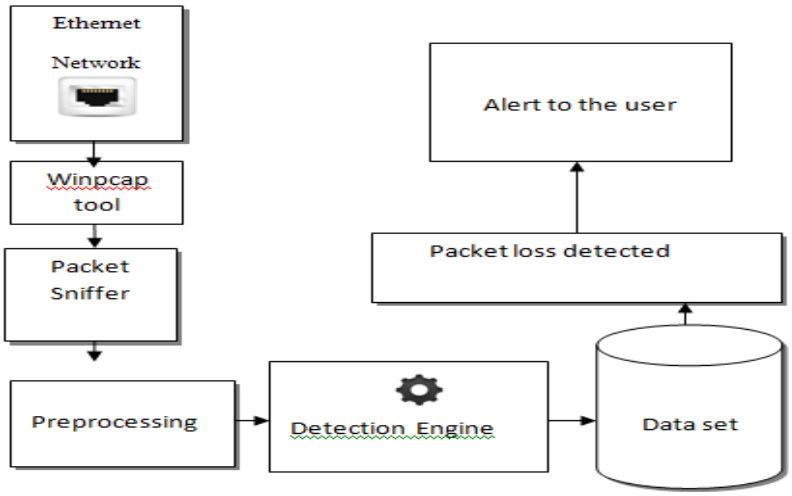

Fig:2 Link error reduction.

4. LITERATURE SURVEY

1) Tao shu and Marwan krunz are interested in determining whether losses are due to link errors only (or) due to the combined effect of link errors and malicious drops. To improve the

detection accuracy, we propose to exploit the correlations between lost packets. To ensure the truthful calculation, they have developed a homomorphic linear authenticator (HLA) based public auditing architecture that allows to verify the truthfulness of the packet loss information reported by nodes. Using this architecture, it requires high computational cost and storage overhead. Persistent packet dropping can effectively degrade the performance of the network.

2) Bobby Sharma Kakoty , S. M. Hazarika , N. Sarma have been discussing about the packet dropping attacks. Malicious nodes invariably drops the packet, when forwarding to destination. A distributed packet dropping attack (PDA) detection methodology named NAODV protocol has been used. With the help of NAODV protocol, it is a nano technology focusing on finding the network performance. The nano technology consists of nano machines, they integrate the AODV routing technique. Using this, we found that it decreases the throughput and increases the packet delay of the network.

3) M. Kiran kumar, A. Sai Harish are interested in detecting the malicious packet dropping attacks in network and to minimize the various security attacks such as black, grey and worm hole. To address this problem they have used router detection protocol that refers to precise number of congestion packet losses. Due to this, we found that collisions and channel errors occurs.

4) P. Papadimitratos and Z. Haas have described, in a open MANET environment, any node can maliciously (or) selfishly disrupt and deny communication of other nodes. They have used secure message transmission protocol which safeguards the data transmission against malicious behavior of another nodes. But it does not achieve end-to-end packet delivery.

5) Venkat Balakrishnan, Vijay Varadharajan, Uday Tupakula, and Phillip Lucs, has been focusing on secure communications among nodes in MANET. To ensure this, trust model known as trust integrated co-operation architecture has been proposed. By using this model, we found that it either fail to protect against flooding attacks or only defend against greedy nodes that drops packets to save battery resources.

6) Haiyun luo, Jiejun kong, Petros Zerfos, Songwu Lu, Lixia Zhang found the lack of a network infrastructure, the dynamics of the network topology and node membership, and the potential attacks from inside the network by malicious and/or noncooperative selfish nodes make the conventional network access control mechanisms not applicable. For this, we use URSA, a ubiquitous and robust access control solution for mobile ad hoc networks. URSA implements ticket certification services through multiple-node consensus and fully localized instantiation. It uses tickets to identify and grant network access to well-behaving nodes. In URSA, no single node monopolizes the access decision or is completely trusted. Instead, multiple nodes jointly monitor a local node and certify/revoke its ticket. By using this, we found that the restricting network access of routing and packet forwarding to well-behaving nodes and denying access from misbehaving nodes are critical for the proper functioning of a mobile ad-hoc network.

7) Sirisha Medidi, Muralidhar Medidi and Sireesh Gavini found that in a MANET, which is prone to security attacks, with node mobility being the primary cause in allowing security. For this purpose an unobtrusive monitoring technique to locate malicious packet drops. Using this makes the network to faults with packets getting misrouted (or) dropped.
8) Kejun Liu, Jing Deng, Pramod K. Varshney, and Kashyap Balakrishnan found the routing misbehaviour in MANET. Due to the open structure and scarcely available battery based energy, node misbehaviours may exists. They have used 2ACK scheme that serves as an add-on technique to detect routing misbehaviour. The idea of 2ACK is to send two-hop ack packets in the opposite direction of routing path. Using this technique, it increases the routing overhead.

5. THE WINPCAP ARCHITECTURE

WinPcap is an architecture[9] for packet capture and network analysis for the Win32 platforms. It includes a kernel-level packet filter, a low-level dynamic link library (packet.dll), and a high-level and systemindependent library (wpcap.dll, based on libpcap version 0.6.2).

The packet filter is a device driver that adds to Windows 95, 98, ME, NT, 2000 and XP the ability to capture and send raw data from a network card, with the possibility to filter and store in a buffer the captured packets[10].
WinPCap can be used to capture the network packets and to generate traffic (sending packets through the network).

Packet.dll is an API that can be used to directly access the functions of the packet driver[12], offering a programming interface independent from Windows.

Wpcap.dll exports a set of high level capture primitives that are compatible with libpcap, the wellknown Unix capture library[13]. These functions allow to capture packets[12] in a way independent from the underlying network hardware and operating system.

WinPCap can capture data sent through current machine (which is used by WinPCap)[11]or to any computer from the LAN (only if a hub or a switch does not direct the traffic - these kinds of equipments tend the packets[12] through the destination computer). Generally, generating the traffic is used to test some software components (capture applications)[13] or hardware components (network adapters, modems).

Under these considerations, WinPcap can be used by different kinds of tools for network analysis (for troubleshooting, security and monitoring):

- Network and protocol analyzers
- Network monitors
- Traffic loggers
- Traffic generators
- Network intrusion detection systems (NIDS)
- Network scanners

□□Security tools

5.1 WinPCap Structure

To capture data from the network, a capture application must directly interact with the network adapter[10]. Therefore the operating system must offer a set of primitives for capture with a view to communicate with the adapter. The purpose of these primitives is to capture network packets transparently from the point of view of the user and to transfer them to the calling application. This part of the kernel should be quick and efficient in order to capture all the packets in real time without losing information.

The ***WinPCap*** Architecture [9]is a hierarchical structure on 3 levels (from the network adapter to an application).

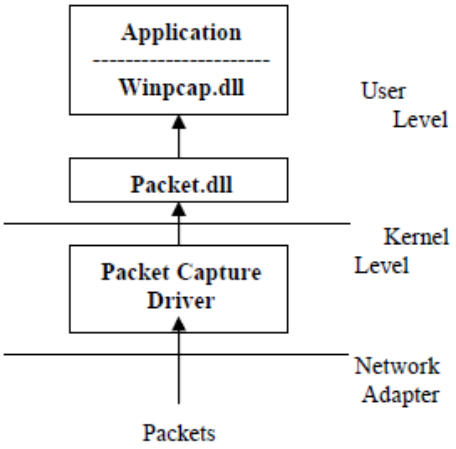

Fig:3 The WinCap Architecture

At the lowest level there is the network adapter. It is used to capture the packets that circulate in the network. It can be set to "promiscuous mode" that means it will accept all the packets even if they are not intended for current computer.

The ***Packet Capture Driver*** is the lowest software level of the capture structure. It is a part of the kernel and interacts with the network adapter to obtain captured packets. It supplies the application a set of functions used to read /write data from the network at data-link level.

Packet.dll works at user level, but it is detached from the capture application.It is a dynamic link library that separates the application from capture driver providing a system-independent capture interface. It allows user's application to be executed on different Windows operating systems without being recompiled. It represents a set of API functions used to access the capture driver directly.

WinPCap.dll (the third level) is a *static* library that is used by the packet captures part of the application. It uses the services exported by Packet.dll, and provides the applications a higher level and a powerful capture interface. It is statically linked; it means it is part of the

application that uses it. The user interface is the highest part of the capture application[13]. It manages the interaction with the user and displays the result of a capture.

A final remark to this description should be useful: the use of term *packet* here may be not very accurate but is most expressive. Capture process is done at Data- Link Layer and the PDU (Protocol Data Units) for this layer is the frame (according to ISO/OSI Model).

5.2 The Format of a WinPCap Application

In WinPcap 3.0,[9] the final stable release of this package, the main improvements worth to be mentioned are:
- kernel buffering rewritten from scratch
- experimental support for remote capture.

The support for SMP machines has been included starting from version 3.0, but only physical interfaces are supported (this is a limitation of Windows and not of WinPcap). All WinPcap can run on the main Win32 operating systems: Windows 95,98,ME, NT4 and 2000, but for Windows XP, WinPcap version 2.3 or higher is required.

The WinPCap[9] action is based on packets capture at the network adapter level. Therefore, any application using WinPCap must follow the steps summarized below:

- Specify the network adapter which will be used for capture.
- Initialize winpcap[10] (mention the network adapter if the capture is on-line).
- Offer a pattern for the captured data (a TCP/IP structure for example), so it can be done a cast at this pattern and obtain significant information (complex applications offer patterns for majority protocols).
- Analyze captured data according with the application type (captured data can be saved and processed off-line).
- Close the capture session.

5.3 The PCapture Application

PCapture[12] is an interactive tool which uses the WinPCap architecture[10] and C/C++ calls for packet captures. Although we used for its interface the MFC (Microsoft Foundation Class) functions, PCapture has a certain degree of independence from the OS, by using the "compatible" calls of WinPCap. Use of threads reduces somehow the predictibility of application behavior only to systems similar to Windows NT, 2000 or XP (we noticed non deterministic behavior during tests on Windows 98 or ME).

The tool is just a preliminary step; it achieves the capture of all packets in teh network and the specification of their type. For "standard" packet types like TCP/IP and UDP the

component fields of the headers are specified, shaped as a tree structure. Going towards the superior level (Application) requires the definition of patterns, corresponding to the types of protocols on this level.

This tool has options for loading and saving capture sessions, using disk files. It is also very simple and friendly from the UI (user interface) point of view. The user can easily select the capture mode, filters, adaptors a.o. An interesting and useful feature proved to be the refreshing option, by which re-filtering of already captured packets[13] is achieved. In other words, although the capture session may be ended, if the number of packets is too big making their analysis too difficult, a new filter can be selected and the screen refreshed accordingly.

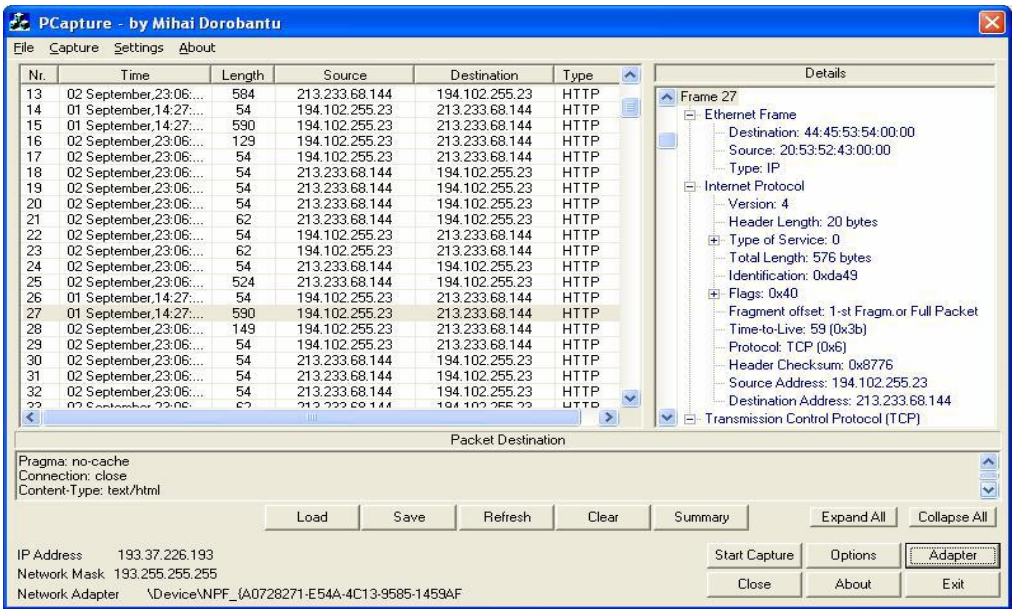

Fig:4 The PCapture Tool

6. CONCLUSION

In this paper, We found that the link errors and malicious packet dropping are the two sources for packet losses in the wireless adhoc network. In order to improve the detection accuracy, we propose the tool called the windows packet capture (winpcap). The purpose of the tool is to give alert to the user about the link errors and the packet drop. And, we will also create an log at the backend, so that the user who request the file, will be given an priority, so based on the priority the system will download the file to the user. Using the clustering algorithm and batch auditing technique we can collect the similar data together and send to the correct user. In the proposed system, we have maintained the three attributes such as the request, proceed and download. The purpose of the log is to make the network a collision and

traffic proof. It stops the multiple request from the user to download a file. Using the tool, it will give alert to the user about the errors and the packet drop which are reported by the nodes.

7. REFERENCES

[1]Tao Shu,Marwan Krunz."Detection of Malicious Packet Dropping in Wireless Ad Hoc Networks Based on Privacy Preserving Public Auditing",Department of Electrical and Computer Engineering,University of Arizona, USA, 2013.

[2]Bobby Sharma Kakoty , S. M. Hazarika , N. Sarma. "NAODV-Distributed Packet Dropping Attack Detection in MANETs ", Dept. of Computer Science & Engineering, School of Engineering, Tezpur University, Tezpur, 2013.

[3]M. Kiran kumar, A. Sai Harish, "A Novel Schema for Detecting Malicious Packet Losses" International Journal of Modern Engineering Research (IJMER) pp-3633-3636 ISSN: 2249-6645, Vol.2, Issue.5, Sep-Oct. 2012.

[4]P. Papadimitratos and Z. Haas," Secure message transmission in mobile ad hoc networks". *Ad Hoc Networks, 1(1):193–209, 2003.*

[5]Venkat Balakrishnan, Vijay Varadharajan, Uday Tupakula, and Phillip Lucs, "Trust Integrated Co-operation Architecture for Mobile Ad-hoc Networks", *Information and Networked Systems Security (INSS) Research, Department of Computing, Macquarie University, Sydney, Australia 2009.*

[6]Haiyun luo, jiejun kong, petros zerfos, songwu lu, lixia zhang, "URSA: Ubiquitous and robust access control for mobile adhoc networks", IEEE transactions on networking, vol 12, no. 6, december 2004.

[7]Sirisha Medidi*, Muralidhar Medidi and Sireesh Gavini , "Detecting Packet Mishandling in Mobile Ad-hoc Networks", School of Electrical Engineering and Computer Science, Washington State University, Pullman 99164-2752.

[8] K. Liu, J. Deng, P. Varshney, and K. Balakrishnan. "An acknowledgement-based approach for the detection of routing misbehavior in MANETs". *IEEE Transactions on Mobile Computing, 6(5):536*–550, May 2006.

[9] WinPCap Home Page – http://winpcap.polito.it

[10]Loris Degioanni, Development of an Architecture for Packet Capture and Network Traffic Analysis, Graduation Thesis, Politecnico Di Torino (Turin, Italy, Mar. 2000), http://winpcap.polito.it/docs/th_degio.zip

[11]Fulvio Risso, Loris Degioanni, An Architecture for High Performance Network Analysis, *Proceedings of the 6th IEEE Symposium on Computers and Communications (ISCC 2001)*, Hammamet, Tunisia, July 2001.

[12]Tim Carstens, Programming with pcap, tutorial, http://broker.dhs.org/pcap.org

[13]Martin Casado, Packet Capture With libpcap and other Low Level Network Tricks, tutorial, http://www.cet.nau.edu/ ~mc8/Socket/Tutorials/section1.html.