Analysis of Effectiveness of CORE Watchdog Mechanism in Mobile Ad Hoc Networks

Jashandeep Kaur Ghuman, Dale Lindskog, Pavol Zavarsky
Information System Security Management
Concordia University College of Alberta (CUCA) Edmonton Canada
jghuman@student.concordia.ab.ca
{dale.lindskog, pavol.zavarsky}@concordia.ab.ca

***Abstract*—This paper explores the effectiveness of CORE-COllaborative REputation Mechanism to combat the node selfishness problem in Mobile Ad Hoc Networks (MANETs) in the situation when the selfish node is aware of CORE as the detection mechanism employed. We first give details of the working of different components of CORE and then analyse the effectiveness of the component called Watchdog (WD) mechanism for two functions namely the DSR Route Discovery and Packet Forwarding in the situation when a node is aware of the mechanism and is trying to undermine the WD on purpose. We enlist certain scenarios that a selfish node could possibly exploit to escape WD detection and be successful at protecting its ability to be selfish in the ad hoc network.**

***Keywords*— mobile ad hoc networks; node selfishness; collaborative mechanisms; CORE; Watchdog mechanism;**

I. INTRODUCTION

Mobile ad hoc networks (MANETs) are self-configuring infrastructure less networks of mobile devices connected by wireless links [1,2]. The devices are free to move in any direction and are capable of performing the roles of both a router and a host. MANETs have a dynamically changing network topology and thus routing updates are frequent. They can be set up geographically anywhere without any external infrastructure support. Designing a protocol for MANETs is a complex task as most of the traditional security methods depend on a fixed infrastructure. The nodes in a MANET are free to join or leave the network and this makes the detection of malicious nodes joining the networks really hard. Also, lack of a centralized architecture leads to absence of no central monitoring agent in the network.

Out of the attacks peculiar to MANETs, node selfishness is a recently discovered passive attack to the network wherein a node refuses to cooperate in the important network activities primarily with the aim of saving energy selfishly for its own purposes. The essential requirement from every node in a MANET to carry out fair share of the essential network functions like network management and packet forwarding and routing increases node sensitivity to selfish behaviour in a MANET scenario where power saving is a major concern. Also, another reason is that the mobile nodes might be possibly owned by selfish users who may want to use the network resources but refuse to make any contribution to the MANET community [4].

Security research in the field of ad hoc networking has seen great advances in the last few years. Selfishness is a recent issue that is gaining importance because the MANET security protocols cannot cope with this problem [2]. Also, selfishness is an insider attack which are always more difficult to deal with than the external attacks [2].

A simulation study done by P. Michiardi and R. Molva shows that if the selfish nodes go undetected at an early stage, they can partition the network and hence severely degrade the network performance [2,3].

There have been various mechanisms proposed to combat the selfishness problem and can be categorized into: i) Virtual Currency-based Systems ii) Reputation based schemes. Certain virtual-currency based mechanisms required special tamperproof hardware to be implemented and others required a centralized server which made its use unrealistic in a real world ad hoc scenario.

The very advantage of 'no requirement' of a temper proof hardware of these schemes is also a disadvantage in itself. The situations where there is no tamperproof hardware or authentication infrastructure, the reliability of important network functions cannot be fully trusted. The basic weakness that exists with the reputation based schemes is these mechanisms cannot securely identify the nodes of the mechanism. Any node can join or leave the network. If the node's reputation value falls below the threshold value where it is deemed 'selfish', the selfish user can simply change its current identity and opt for leaving the network and then re-joining the network pretending to be a 'new' user and start from the very beginning. This is very much possible as according to [2], each agent's identity is normally a pseudonym on most online reputation systems and pseudonym can be changed easily [8,16,17].

In spite of the basic disadvantage, out of the two schemes, the reputation mechanisms offer a feasible solution for MANETs [2].

Thus research efforts in this field are continuously evolving so as to come up with better ways to detect and ultimately isolate the selfish nodes. The reputation based schemes are based on the idea that MANET nodes are like the members of a community that share a common resource and exhibit cooperative behaviour to use the resources. The members are often unrelated to each other and the 'reputation' of the members plays an important role for most of other members to decide whether a specific member is cooperative or selfish.

There are various mechanisms proposed that fall under the reputation based schemes namely CONFIDANT and CORE. CONFIDANT is the acronym that stands for "Cooperation of Nodes, Fairness In Dynamic Ad-Hoc NeTworks". This is the

term coined by Buchegger and Le Boudec in their paper [6]. The approach followed is that it detects malicious nodes by means of observations. Reputation is the term for evaluating routing and forwarding behavior according to the protocol. Trust is the term for evaluation of participation of a node. Limitation - the basic approach of CONFIDANT has value only if each node's identity is persistent, else it can be vulnerable to spoofing attacks [2,6]. Another limitation is that it is vulnerable to the problem of spreading wrong accusations [2,6].

CORE or Collaborative Reputation mechanism seems to offer a better solution to the selfishness problem than CONFIDANT. Firstly, certain formulae in the mechanism itself are aimed at minimizing the problems due to false detection of a node's misbehaviour. The subjective reputation gathered in the mechanism takes into account the past observations. This is an advantage in cases when the nodes are mistakenly assumed to be 'selfish' when the sole reason might be that they could not perform the required function because of some other reason, for instance a link breakage in the network. So the sporadic misbehaviour of the node is not given much importance.

Secondly, unlike CONFIDANT, CORE solves the problem of maliciously spreading wrong information about the other nodes of the network in order to lower their reputation value. The advantage of this feature is that denial-of-service (DoS) attacks that could be caused by broadcast of negative ratings by the network nodes, are prevented [7,9]. However, it does suffer the problem that is basic to reputation based schemes: a selfish node could change its identity and thus get rid of its bad reputation.

This paper is aimed at analysing the effectiveness of the CORE mechanism in the situation when a selfish node is aware of CORE as the detection mechanism employed and wants to stay selfish in the network without being detected by the mechanism.

This paper is organized as follows: Section II describes the CORE mechanism. Section III gives details of the Watchdog mechanism. Section IV analyses the effectiveness of Watchdog (WD) component of CORE mechanism when a node is trying to undermine the WD on purpose. Finally Section V concludes.

II. CORE MECHANISM DETAILS

COllaborative REputation or CORE mechanism is a generic mechanism that is possible to extend with basic network functions like network management, packet forwarding etc. [7]. CORE is based on Dynamic Source Routing (DSR) protocol which is an 'on demand' protocol. It allows a node to 'dynamically' discover a route to any other node in the network for which it has no path [7,13].

CORE uses reputation scheme in association with the collaborative monitoring technique [2,7]. Every member of the network uses reputation technique to store information about an entity's contribution to the network operations [7,9]. Reputation is basically a measure of the rate of collaboration of an entity [7,8] and is fundamentally a social concept [16].

The reputation information in CORE is gathered in the following ways:

1) *Subjective Reputation*: Calculated when a node directly observes the other node's behaviour. It gives more relevance to the past observations.
2) *Indirect Reputation*: Information provided by other members of the community.
3) *Functional Reputation*: This is basically the subjective and indirect reputation calculated with respect to different functions f.

III. THE WATCHDOG MECHANISM

The reputation table updates its entries based on the Watchdog mechanism. It is the component of CORE which, based on the feedback information, implements the validation phase by observing if the expected $e_r(f)$ and observed results $o_r(f)$ coincide or not.

If a node needs to monitor the correct execution of a function in the neighbouring node, it activates the WD specific to that function. The CORE WD relies on the promiscuous mode operation. It suffers from certain inherent weaknesses which are that it might not be able to detect a misbehaving node in situations of: 1) receiver collisions, 2) ambiguous collisions, 3) partial dropping, 4) false misbehaviour, 5) collusion and 6) limited transmission power [7].

Requester and *Provider* are the two protocol entities in CORE. Requester is the entity that requests for the execution of a specific function 'f' and the entity that executes 'f' is called Provider.

The CORE protocol execution occurs in the following ways:

1) *No misbehaviour detected*: As the requested function was correctly executed, the requestor disarms the WD. RT is not updated in this case.
2) *Misbehaviour detected*: Corresponding entry for the misbehaving node will be updated in the RT; the reputation value related to the misbehaving entity is decreased.
3) *Request made by a misbehaving entity*: The reputation value for the requester is checked in the global RT and the requested function is not executed by the provider if the reputation value is negative.

IV. WATCHDOG EFFECTIVENESS ANALYSIS

In this section, we attempt at enlisting few scenarios, apart from the inherent weaknesses, which a WD might not be able to detect and that could possibly make plausible alterations in the RT. Thomas H. Ptacek and Timothy N. Newsham list certain insertion attacks in the paper "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection" that exploit the passive protocol analysis mechanism of data collection which basically is to sniff the network data unobtrusively and then carefully observe it to find out patterns of any suspicious activity. Since the WD in CORE is

essentially based on passive protocol analysis, most of insertion attacks given in the paper hold true for CORE WD.

We analyse the WD effectiveness, under the circumstances that a node is aware of the detection mechanism, for two basic network functions namely the DSR Route Discovery and Packet Forwarding.

A. *Functions Details*

DSR Route Discovery: The CORE protocol can be thought of as a layer on top of the DSR protocol, and the function f that has to be monitored corresponds to the Route Discovery function of the DSR protocol [10]. When a source node (A) wishes to communicate with a destination node (B), the Route Discovery function is invoked by it as it does not know the path to B. It broadcasts a ROUTE_REQUEST packet to all its neighbours that contain the address of the destination. The neighbouring nodes append their own addresses to the ROUTE_REQUEST packet and then they re-broadcast it. This process continues till the ROUTE_REQUEST packet reaches the destination node. Destination node then sends a ROUTE_REPLY packet to inform the source node of the discovered path by reversing the route path or using the route to the source node if it has one in its route cache.

Packet Forwarding: A node can send packets to the destination node once it has obtained a route for the same using the DSR Route Discovery function. All the nodes in the path will have to then perform the PF function in order to be deemed cooperative nodes.

B. *Insertion attacks on IP datagram*

Most of these insertion attacks can be generalised for both the DSR Route Discovery and the PF functions for an IP datagram.

When the selfish node is aware of the CORE mechanism details, and knows that it is under observation by the WD, it probably will not indulge in the actions of *not* forwarding the packets because it would lead to its ultimate detection by the WD. But it definitely knows that it would have to exhibit only the *expected* behaviour to achieve expected results. And the expected result in CORE mechanism corresponds to the correct execution of the function *as monitored by the WD* [7]. Thus, with the insertion attacks, if a selfish node can be successful in fooling the WD into thinking that something is (or is not) happening on the network [25], it could possibly find out a way that could be used to stay selfish in the network without being detected. Though this behaviour involves some 'malicious' actions by the node initially, the key objective in doing so is *not* to launch an attack on the network but just to save itself from doing much work in the long run.

There are at least two ways in which an insertion attack could work:

1) *The final destination does not receive the packet:* The selfish node could resort to this method, for instance in DSR Route Discovery process, in order to undermine the ability of the source node to establish a route to the destination node through itself. Reason being that if the route is established that involves the selfish node as one of the members of the path, it would then have to perform the important network functions like packet forwarding.

One of the methods to carry this out is with the help of the TTL (time to live) field of the IP packet. The TTL field tells the number of "hops"' a packet is allowed on its way to the destination node. The TTL value is decremented every time a packet is forwarded by a router and the packet is thus ultimately dropped when the TTL runs out.

A selfish node could set the TTL value too short for the packet to actually arrive at its destination [25], and then forward the packet. By doing so, it can make the WD into thinking that there was no misbehaviour as it *did* forward the packet. Unless the WD examines the TTL field, and can determine the consequences of that low value, the WD will see it as a legitimate effort to assist that packet to arrive at its final destination.

Another way is that a selfish node could encapsulate the IP packet in an invalid MAC address and this would result in the end host not receiving the packet.

If the node is aware of the link-layer address of the WD, it could address the fake packets to the WD without ever allowing the host specified as the IP destination to see the packet [25].

2) *The final destination receives, but does not process the packet:* If the entity knows that the WD is concerned only about packet forwarding and not about the bad packets, it can revert to this method to send packets to an end-system that it will reject, but that the WD will think are valid [25]. If it is successful in doing so, it has behaved legitimately in front of the WD but in reality, it has taken steps to be selfish, which in this case would be to indulge in forwarding far less traffic than it normally would when it is exhibiting purely cooperative behaviour. Thus in this case, the network entity wants to save itself from doing maximum work and is too selfish for participating in the essential network functions.

One way to establish this is to have an invalid header field in the IP datagram. For instance, assigning a value other than 4 to the 'version' field will prevent the packet from being routed altogether.

Having a bad checksum for the IP datagram will make an end system reject the packet because it will not be processed by most IP implementations. Unless the WD checks the accuracy of checksum on every packet which may actually seem unnecessary [25], the destination node will be made to simply discard the packet altogether.

C. *Other Attacks*

Most of the insertion attacks could potentially be resolved if the WD is rebuilt fundamentally in a way to observe all the fields cited in the above scenarios. But the problem does not end there, because modifying the IP datagrams to carry out insertion attacks is not the only method that a selfish node can exploit to stay selfish in the long run.

There are other ways for the selfish node to achieve the same results for itself. An example is of an option called 'timestamp' which requests the placement of a timestamp within the packet by certain recipients of the datagram [25]. There is a code that processes the timestamp option. This code can be forced to discard the packet (if the option is malformed) [25].

If a selfish node has details on the operating system of the destination node, it could configure it to automatically reject the source routed packets. The end-system's configuration information such as the operating system running on it and its versions might be helpful for the WD in determining whether the packet will be accepted by the system at all because one operating system might process a packet differently than another. The problem here is that the WD tries to detect selfish behaviour by carefully observing the information available in the packets that it captures by the passive protocol analysis method. This method certainly cannot provide the WD with any important information about the end-system configuration.

If reliable source of information can be made available for the WD, most ambiguities might be solved. But the problem is that the basic functioning of ad hoc networks makes the collection of this information difficult. For instance information regarding topology of the network might solve most of the ambiguities for the WD but this information cannot be obtained reliably for a MANET because the network topology is always changing as the node have high rate of mobility.

It is thus not unreasonable to say that building a sophisticated WD for CORE mechanism to detect every possible misbehaviour on the network is highly unlikely.

D. *Selective Selfishness*

Apart from all the malicious attempts stated above, a node can simply choose to be 'selectively' selfish and try different methods without the fear of being detected by the WD. This is because in CORE, the reputation value calculated for every node is compositional and a simple one time selfish behaviour will not lead to its ultimate isolation from the network. Since the selfish node is aware of CORE details, it knows that even if it is caught, the only downside it possibly will suffer for its misbehaviour is that its reputation value will be decremented but only by a small fraction and it always has a chance to redeem itself. So, it can experiment with new attacks unless it finds one that works the best for it. It can choose to be selfish or cooperative for different instances and thus stay selectively selfish as long as it wants to stay selfish.

Thus this basic approach of CORE of 'non isolation' on a mere one time misbehaviour together with the problem of absence of a reliable source of information opens the door for endless possibilities for a selfish node to successfully stay selfish in the network as long as it wants to be. Hence building a sophisticated WD for CORE is really difficult.

V. CONCLUSION

When a node is trying to undermine the CORE Watchdog (WD) mechanism on purpose, a reliable WD is really difficult to obtain. There are endless possibilities for a selfish node to exploit and it is almost impossible to configure a sophisticated WD mechanism that cannot be evaded or that could completely detect every possible misbehaviour a node could indulge in to stay selfish in the network.

CORE is a good mechanism only in the case of security-by-obscurity. If the selfish node is aware of the mechanism, it is not the best mechanism to be employed to detect a selfish node.

REFERENCES

[1] Information Technologies Library Advanced Network Topologies Division, NIST. Mobile Ad Hoc Networks. Available: http://www.antd.nist.gov/wahn_mahn.shtml

[2] S. Basagni et. al.,"Ad Hoc Networks Security" in Mobile Ad Hoc Networking, 2004, John Wiley & Sons Inc. Pub., pp: 329-347.

[3] P. Michiardi and R. Molva, "Simulation-based analysis of Security Exposures in Mobile Ad Hoc Networks", In Proceedings of European Wireless Conference, 2002.

[4] J.N. Al-Karaki and A.E. Kamal, "Stimulating Node Cooperation in Mobile Ad Hoc Networks".Wireless Pers Commun (2008), p: 219-239.

[5] L. Buttyan and J.-P. Hubaux, "Nuglets: A Virtual Currency to Stimulate Cooperation in Self Organized Ad Hoc Networks". Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne, 2001.

[6] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes: Fairness in Dynamic Ad Hoc Networks", in Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH, June, 2002.

[7] P. Michiardi and R. Molva, "CORE: A COllaborative REputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks", IFIP-Communicatin and Multimedia Securtiy Conference, 2002.

[8] J. Liu and V. Issarny, "Enhanced Reputation Mechanism for Mobile Ad Hoc Networks", Second International Conference on Trust Management: iTrust, 2004, pp: 48-62.

[9] J.N. Al-Karaki and A.E. Kamal, "Stimulating Node Cooperation in Mobile Ad Hoc Networks", Wireless Personal Communications: An International Journal, Volume 44 Issue 2, 2008, pp: 219-239.

[10] P. Michiardi and R. Molva, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad Hoc Networks". Institut Eurecom Research Report RR-02-070, April 2002.

[11] S. Buchegger and J.Y. Le Boudec, "Effect of Rumor Spreading in Reputation Systems for Mobile Ad-Hoc Networks", in Proceedings of WiOpt, Modelling and Optimisation in Mobile Ad Hoc and Wireless Networks, 2003.

[12] M. A. Ali and Y. Sarwar, "Security Issues regarding MANET (Mobile Ad Hoc Networks: Challenges and Solutions", Master Thesis, Dept. of Comp. Sc., Berkling Inst. of Technology, Karlskrona, Sweden, 2011.

[13] D. B. Johnson et.al ,"DSR: The Dynamic Source Routing Protocol for Multi Hop Wireless Networks". [Online] Available: http://www.ietf.org/rfc/rfc4728.txt

[14] Frank Stajano and Ross Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks", Proceedings of the 7th Security Protocols Workshop, 1999, pp: 172-194.

[15] S. Marti, T. Giuli, K. Lai, and M. Baker,"Mitigating Routing Misbehavior in Mobile ad hoc Networks". In Proceedings of International Conference on Mobile Computing and Networking, (MOBICOM), 2000, pp: 255-265.

[16] L. Mui, M. Mohtashemi and A. Halberstadt, "A Computational Model of Trust and Reputation", in Proceedings of the 35th Hawaii International Conference on System Science (HICSS), 2002.

[17] G. Zacharia and P. Maes, "Trust Management Through Reputation Mechanisms", Applied Artificial Intelligence, 2000, pp:881-907.

[18] A. E. Gamal, J. Mammen, B. Prabhakar, D Shah, "Throughput-Delay Trade-off in Wireless Networks", 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies, INFOCOM, Volume 1, 2004

[19] Y.-C. Hu, A. Perrig and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks", in Proceedings of MOBICOM 2.

[20] H. Yang et. al, "Security in Mobile Ad Hoc /Networks: Challenges and Solutions". IEEE Wireless Communications, Volume 1, Issue 1, 2004, pp: 38-47.

[21] R. L. Mathew and Prof. P. Petchimuthu, "Detecting Selfish Nodes in MANETs Using Collaborative Watchdogs", International Journal of

Advanced Research in Computer Science and Software Engineering Research Paper, 2013.

[22] J. Hortelano et. al. "Watchdog intrusion detection systems: Are they feasible in manets?," in XXI Jornadas de Paralelismo (CEDI'2010), 2010.

[23] Dr. S.C. Sharma and G.S. Mamatha "Network Layer attacks and defense mechanisms in MANETS- A Survey". International Journal of Computer Applications (0975-8887), Volume 9 – No. 9, 2010.

[24] S Krishna and A.L. Vallikannu, "Detecting Malicious Nodes For Secure Routing in MANETS Using Reputation Based Mechanism", International Journal of Scientific & Engineering Research, Volume 1, Issue 3, December-2010 1

[25] T. H. Ptacek, and T. N. Newsham. "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", Technical Report, Secure Networks Inc., 1998.

[26] L. Xu et. al. "Analysis and countermeasures of selfish node problem in mobile ad hoc network," in Proceedings of the Tenth International Conference on Computer Supported Cooperative Work in Design (CSCWD '06), 2006.

[27] Y. Cho et. al.,"Insider Threats against Trust Mechanism with Watchdog and Defending Approaches in Wireless Sensor Networks", Security and Privacy Workshops (SPW), IEEE Symposioum, 2012, pp.134-141.