Dealing with Vampire Attacks: A Survey

Mrs. Damayanti D. Pawar [*], Ms. Megha Singh [**]

[*] Post Graduate Student, C.I.I.T, Indore
[**] Assistant Professor, C.I.I.T, Indore

Abstract- Wireless sensor networks are the widely used networks in case of research and persuasive computing. These networks consist of number of sensor nodes which collect data and send it to the data aggregator for further processing. The use of low powered and energy efficient devices for the collection of data is one of the requirements of these networks and thus energy management becomes an important part of them. One of the major issues related to security which is faced by these networks is that of vampire attacks. These attacks are launched by the adversaries in order to drain the energy from the network so as to create denial of services and abruption. This paper focuses on vampire attacks, their effects on wires less sensor networks and finally also discusses about the modified clean state sensor network routing protocol which is one of the promising work to resist and protect again the vampire attacks.

Index Terms- Vampire Attacks, Denial of Service, Wireless adhoc networks.

General Terms - Network Security

I. Introduction

Vampire attacks are widely spreading attacks in the area of wireless adhoc networks which focus on depletion of resources specially the power of the network. Also these attacks are not protocol specific and hence many of the protocols can be easily attacked by them. This paper studies the effect of the vampire attacks on different protocols and also investigates about the different measures that can be adopted to reduce their effect.

II. Background

A wireless sensor network is built up of nodes where each node is connected to one or more sensors. One of the requirements of the wireless adhoc network is that of low energy use. The reason behind this is that the sensors may be placed in remote area and service of node may not be possible. Hence life time of the node is determined by battery life of the node. Energy being a crucial part of the wireless adhoc networks, the vampire attacks focus on draining of the same from the networks so as to initiate and increase the denial of service attacks. Many protocols like link state, distance vector, source routing, geographical and beacon are affected by the vampire attacks. The paper focuses on study of type of attacks and countermeasures for preventing them. It also discusses about the clean state sensor networking protocol which is found to give a better protection against the vampire attacks.

III. Issues

Wireless adhoc networks or wireless sensor networks are the networks consisting of low power, small size devices called as nodes which can sense the environment and collect data and then this data can be communicated using wireless links. These types of networks are widely used in applications like space research, military, civilian and scientific researches. In such applications, the life of network plays a crucial role. This life of the network can be exhausted by vampire attacks. Vampire is a node that has been compromised in Wireless sensor network by an adversary. Through this node adversary can send protocol compliant messages to other nodes continuously and let them answer the messages so as to ensure that each node in the network looses energy faster (energy depletion) causing the failure of the whole network soon. Vasserman and Hopper explored resource depletion attacks in ad hoc sensor networks. They studied all routing protocols and found that vampire attacks can target any routing protocol. Vampire attack can also be described as the composition and transmission of a message that causes more energy to be consumed by the network than if an uncompromised node would have transmitted the same message. This effect is introduced by the adversary by changing the packet headers.

IV. Vulnerabilities

Vampire attacks are not protocol specific. They explore the vulnerabilities of the protocol and use them for starting the attack. Also they do not flood the networks and hence are difficult to prevent and detect. Their main aim is to transmit as fewer amounts of data as possible and retain most of the data and processing at the node so that its energy gets exhausted. Some of the vulnerabilities that are explored by these attacks are:

- In case of source routing where the paths are already decided at the source, the adversary can create malicious packet source to describe paths that are longer than optimal so as to waste the energy at intermediate nodes who will forward the packets as mentioned in the source route.
- In case of independent routing where every node takes the forwarding decision independently, the adversary can make use of directional antenna and wormhole attacks to deliver the packets to multiple remote network positions to force them to process the packet when in reality they should not even have received the packet. This increases the network wide energy consumption thus draining the life of network.
- One more vulnerability that can be explored by the adversary is that instead of attacking in packet

forwarding phase, it can also attack the route and topology discovery phases. If the discovery messages are flooded, then it can consume the energy at every node just at the cost of a single message.

Thus protection of wireless sensor networks from vampire attacks is one of the challenges being faced recently.

V. Organization of the Paper

Section 1 gives an introduction to the concept of vampire attacks in wireless sensor networks.. Section 2 describes the protocols affected by vampire attacks. Section 3 focuses on the types of attacks. Section 4 briefly describes the modified clean state sensor network routing and finally the paper is concluded in Section 5.

VI. Targeted Protocols

This section discusses about the protocols that can be affected by the vampire attacks. There are basically two types of protocols used for routing purpose in wireless sensor networks

- Stateful protocols: The stateful ad hoc routing protocols require node to maintain some routing information that is collected using the routing protocol (e.g., through route request propagation or by reversing paths taken by the query). Stateful routing protocols need the routing information maintained at each intermediate node through the data forwarding path.

Example: DSR (Dynamic state routing protocol), OLSR (Optimized Link State Routing Protocol), DSDV (Destination-Sequenced Distance-Vector Routing)

- Stateless Protocols: These kinds of protocols only track the position of their neighbors and select among them a neighbor that is likely to be closer to the destination. Stateful routing may not be efficient or even possible for very large networks with limited sensor node capabilities. Accordingly, stateless routing protocols which do not maintain per-route state have been proposed. They scale effectively in terms of routing overhead because the tracked routing information does not grow with the network size or the number of active sinks.

Example: Geographic forwarding protocols.

The protocols that are widely used in wireless sensor networks and which are found to be prone to vampire attacks are described below

- DSR (Dynamic State Routing Protocol): It is a source routing protocol where the sender of the packet specifies the route that the packet should take through the network. It specifies either a partial or a complete route. It consists of two major phases of route discovery and route maintenance. When the source node wants to send a packet to a destination, it looks up its route cache to determine if it already contains a route to the destination. Each node maintains route caches containing the source routes that it is aware of. The node updates entries in the route cache as and when it learns about new routes. If the node finds that an unexpired route to the destination exists, then it uses this route to send the packet. On the other hand, if the node does not have such a route, then it initiates the route discovery process by broadcasting a route request packet throughout the network. The route request packet contains the address (usually the IP) of the source and the destination, and a unique identification number. Each intermediate node checks whether it knows of a route to the destination. If it does not, it appends its address to the route record of the packet and forwards the packet to its neighbors. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and it's address is not present in the route record of the packet. A route reply is generated when either the destination or an intermediate node with current information about the destination receives the route request packet. A route request packet reaching such a node already contains, in its route record, the sequence of hops taken from the source to this node. As the route request packet propagates through the network, the route record is formed.
- OLSR (Optimized Link State Routing Protocol): It is a proactive protocol designed for mobile adhoc networks. It performs hop by hop routing that is each node uses its most recent information to route a packet. It uses hello and topology control (TC) messages to discover and then disseminate link state information throughout the mobile ad hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths.
- DSDV (Destination-Sequenced Distance-Vector Routing) Protocol: It is a table-driven routing scheme for ad hoc mobile networks based on the Bellman–Ford algorithm. It solves the routing loop problem. Each entry in the routing table contains a sequence number, the sequence numbers are generally even if a link is present; else, an odd number is used. The number is generated by the destination, and the emitter needs to send out the next update with this number. Routing information is distributed between nodes by sending full dumps infrequently and smaller incremental updates more frequently
- GPSR (Greedy perimeter Stateless Routing) Protocol: is a responsive and efficient routing protocol for mobile, wireless networks. Unlike established routing algorithms before it, which use graph-theoretic notions of shortest paths and transitive reachability to find routes, GPSR exploits the correspondence between geographic position and connectivity in a wireless network, by using the positions of nodes to make packet forwarding decisions. GPSR uses greedy forwarding to forward packets to nodes that are always progressively closer to the destination. In regions of the network where such a greedy path does not exist (i.e., the only path requires that one move temporarily farther away from the destination), GPSR recovers by forwarding in perimeter mode, in which a packet

traverses successively closer faces of a planar subgraph of the full radio network connectivity graph, until reaching a node closer to the destination, where greedy forwarding resumes.

- BVR (Beacon Vector Routing) Protocol: BVR defines a set of coordinates and a distance function to enable scalable greedy forwarding. These coordinates are defined in reference to a set of beacons which are a small set of randomly chosen nodes; using a fairly standard reverse path tree construction algorithm every node learns its distance, in hops, to each of the beacons. A node's coordinates is a vector of these distances. On the occasion that greedy routing with these coordinates fails, a correction mechanism is used that guarantees delivery

VII. Attacks and Counter Measures

This section discusses about the attacks and the counter measures taken by the algorithm to mitigate the attacks.

- Carousal attack: In this type of attack, an adversary sends a packet with route composed as a series of loops, such that the same node appears in the route many times. This increases the route length and thus the energy consumption. The position of adversary plays an important role in the success of this attack. In case of DSR, the carousel attack can be prevented entirely by having forwarding nodes check source routes for loops. While this adds extra forwarding logic and thus more overhead, the gain can be expected to be worthwhile in malicious environments. When a loop is detected, the source route could be corrected and the packet sent on, but one of the attractive features of source routing is that the route can itself be signed by the source. Therefore, it is better to simply drop the packet, especially considering that the sending node is likely malicious (honest nodes should not introduce loops). An alternate solution is to alter how intermediate nodes process the source route. To forward a message, a node must determine the next hop by locating itself in the source route. If a node searches for itself from the destination backward instead from the source forward, any loop that includes the current node will be automatically truncated (the last instance of the local node will be found in the source route rather than the first). No extra processing is required for this defense, since a node must perform this check anyway — we only alter the way the check is done.
- Stretch attack: It is where a malicious node constructs artificially long source routes, causing packets to traverse a larger route than optimal number of nodes and draining extra energy. Like for example an honest node would select 4 nodes for transmission of packet whereas an adversary may select all the nodes for packet transmission thereby increasing the route length. The stretch attack is more challenging to prevent. Its success rests on the forwarding node not checking for optimality of the route. If we call the no-optimization case "strict" source routing, since the route is followed exactly as specified in the header, we can define loose source routing, where intermediate nodes may replace part or the entire route in the packet header if they know of a better route to the destination. This makes it necessary for nodes to discover and cache optimal routes to at least some fraction of other nodes, partially defeating the as-needed discovery advantage. Moreover, caching must be done carefully lest a maliciously suboptimal route be introduced.

Thus DSR is more prone to above mentioned attacks.

In case of stateful protocols, routes in link-state and distance-vector networks are built dynamically from many independent forwarding decisions, so adversaries have limited power to affect packet forwarding, making these protocols immune to carousel and stretch attacks. In fact, any time adversaries cannot specify the full path, the potential for Vampire attack is reduced. However, malicious nodes can still mis-forward packets, forcing packet forwarding by nodes that would not normally be along packet paths. Same scenario is prevalent in case of coordinate and beacon based protocols. Such attacks are described next.

- Directional antenna attack: In a directional antenna adversaries can deposit a packet in arbitrary parts of the network, while also forwarding the packet locally. This consumes the energy of nodes that would not have had to process the original packet, with the expected additional honest energy expenditure of O(d), where d is the network diameter, making d /2 the expected length of the path to an arbitrary destination from the furthest point in the network. This attack can be considered a half-wormhole attack, since a directional antenna constitutes a private communication channel, but the node on the other end is not necessarily malicious. It can be performed more than once, depositing the packet at various distant points in the network, at the additional cost to the adversary for each use of the directional antenna.
- Malicious discovery attack: It is also called as spurious route discovery. In most protocols, every node will forward route discovery packets (and sometimes route responses as well), meaning it is possible to initiate a flood by sending a single message. Systems that perform as-needed route discovery, such as AODV and DSR, are particularly vulnerable, since nodes may legitimately initiate discovery at any time, not just during a topology change. A malicious node has a number of ways to induce a perceived topology change: it may simply falsely claim that a link is down, or claim a new link to a non-existent node. However, nearby nodes might be able to monitor communication to detect link failure (using some kind of neighborhood update scheme). Still, short route failures can be safely ignored in networks of sufficient density. More serious attacks become possible when nodes claim that a long distance route has changed. This attack is trivial in open networks with unauthenticated routes, since a single node can emulate multiple nodes in neighbor

relationships or falsely claim nodes as neighbors. Here, two cooperating adversaries communicating through a wormhole can repeatedly announce and withdraw routes that use this wormhole, causing a theoretical energy usage increase of a factor of O(N) per packet. Adding more malicious nodes to the mix increases the number of possible route announce/withdrawal pairs.

To handle the above problems, one of the recently proposed solutions is that of clean state sensor network routing.

VIII. Clean State Sensor Network Routing

This section discusses about the modified clean state sensor network routing protocol described by Vasserman et.al.
Steps of working are as follows:

1. Topology Discovery Phase: It is repeated on a fixed schedule to ensure that topology information stays current. Discovery deterministically organizes nodes into a tree that will later be used as an addressing scheme. When discovery begins, each node has a limited view of the network — the node knows only itself. Nodes discover their neighbors using local broadcast, and form ever-expanding "neighborhoods," stopping when the entire network is a single group. Throughout this process, nodes build a tree of neighbor relationships and group membership that will later be used for addressing and routing. At the end of discovery, each node should compute the same address tree as other nodes. All leaf nodes in the tree are physical nodes in the network, and their virtual addresses correspond to their position in the tree. All nodes learn each others' virtual addresses and cryptographic keys. The final address tree is verifiable after network convergence, and all forwarding decisions can be independently verified. Furthermore, assuming each legitimate network node has a unique certificate of membership (assigned before network deployment), nodes who attempt to join multiple groups, produce clones of themselves in multiple locations, or otherwise cheat during discovery can be identified and evicted.
2. Topology Discovery: Discovery begins with a time-limited period during which every node must announce its presence by broadcasting a certificate of identity, including its public key (from now on referred to as node ID), signed by a trusted offline authority. Each node starts as its own group of size one, with a virtual address 0. Nodes who over hear presence broadcasts form groups with their neighbors. When two individual nodes (each with an initial address 0) form a group of size two, one of them takes the address 0, and the other becomes 1. Groups merge preferentially with the smallest neighboring group, which may be a single node. We may think of groups acting as individual nodes, with decisions made using secure multiparty computation. Like individual nodes, each group will initially choose a group address 0, and will choose 0 or 1 when merging with another group. Each group member prepends the group address to their own address, e.g. node 0 in group 0 becomes 0.0, and node 0 in group 1 becomes 1.0, and so on. Each time two groups merge, the address of each node is lengthened by one bit. Implicitly, this forms a binary tree of all addresses in the network, with node addresses as leaved. Note that this tree is not a virtual coordinate system, as the only information coded by the tree are neighbor relationships among nodes. Nodes will request to join with the smallest group in their vicinity, with ties broken by group IDs, which are computed cooperatively by the entire group as a deterministic function of individual member IDs. When larger groups merge, they both broadcast their group IDs (and the IDs of all group members) to each other, and proceed with a merge protocol identical to the two-node case. Groups that have grown large enough that some members are not within radio range of other groups will communicate through "gateway nodes," which are within range of both groups. Each node stores the identity of one or more nodes through which it heard an announcement that another group exists. That node may have itself heard the information second-hand, so every node within a group will end up with a next-hop path to every other group, as in distance-vector. Topology discovery proceeds in this manner until all network nodes are members of a single group. By the end of topology discovery, each node learns every other node's virtual address, public key, and certificate, since every group members knows the identities of all other group members and the network converges to a single group

Packet Forwarding Phase: During the forwarding phase, all decisions are made independently by each node. When receiving a packet, a node determines the next hop by finding the most significant bit of its address that differs from the message originator's address. Thus every forwarding event (except when a packet is moving within a group in order to reach a gateway node to proceed to the next group) shortens the logical distance to the destination, since node addresses should be strictly closer to the destination. To resist against the vampire attacks this phase plays an important role. The property of no-backtracking must be preserved in order to resist the vampire attacks. This property of No-backtracking is satisfied if every packet p traverses the same number of hops whether or not an adversary is present in the network. Thus it focuses on not allowing the packet to deviate from its destination too much. To preserve no-backtracking, a verifiable path history is added to every PLGP packet, similar to route authentications and path-vector signatures. This packet history is used together with PLGP's tree routing structure so that every node can securely verify progress, preventing any significant adversarial influence on the path taken by any packet which traverses at least one honest node. Whenever node n forwards packet p, it this by attaching a non-replayable attestation (signature). These signatures form a chain attached to every packet, allowing any node receiving it to validate its path. Every forwarding node verifies the attestation chain to ensure that the packet has never travelled away from its destination in the logical address space. This keeps the adversaries away from packet spoofing. Also as all messages are signed by their

originator, messages from honest nodes cannot be arbitrarily modified by malicious nodes wishing to remain undetected. Rather, the adversary can only alter packet fields that are changed en route (and so are not authenticated), so only the route attestation field can be altered, shortened, or removed entirely. To prevent truncation, which would allow Vampires to hide the fact that they are moving a packet away from its destination, one-way signature chain construction is used, which allow nodes to add links to an existing signature chain, but not remove links, making attestations append-only.

Thus the protocol provably bounds the ratio of energy used in the adversarial scenario to that used with only honest nodes to 1, and thus resists Vampire attacks. This is achieved because packet progress is securely verifiable.

IX. CONCLUSION

In this paper we discuss about the wireless sensor networks and the vulnerabilities present in them. We also discussed about the vampire attacks and their types. We then investigated the different protocols that are used in the wireless sensor networks and how they get affected by the vampire attacks. Also it was studied that out of the solutions available in the literature, a modified version of clean state secure network routing protocol is available which employs an effective packet forwarding strategy so as to keep the adversaries away from draining the energy of the network.

REFERENCES

[1] Eugene Y. Vasserman, Nicholas Hopper- Vampire attacks: Draining life from wireless Ad-hoc Sensor networks IEEE TRANSACTIONS ON MOBILE COMPUTING VOL.12 NO.2 YEAR 2014

[2] Thomas H. Clausen and Philippe Jacquet, Optimized link state routing protocol (OLSR), 2003.

[3] David B. Johnson, David A. Maltz, and Josh Broch, DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, Ad hoc networking, 2001.

[4] Bryan Parno, Mark Luk, Evan Gaustad, and Adrian Perrig, Secure sensornetwork routing: A clean-slate approach, CoNEXT, 2006.

[5] Chris Karlof and David Wagner, Secure routing in wireless sensor networks: attacks and countermeasures, IEEE international workshop on sensor network protocols and applications, 2003.

[6] Charles E. Perkins and Pravin Bhagwat, Highly dynamic desination sequenced distance-vector routing (DSDV) for mobile computers, Conference on communications architectures, protocols and applications, 1994

AUTHORS

First Author – Mrs. Damayanti D. Pawar, MTech(2nd Year), C.I.I.T, Indore, damayantipawar09@gmail.com
Second Author – Ms. Megha Singh, Assistant Professor, C.I.I.T., Indore, maggi.megha@gmail.com
Third Author – Mrs. Damayanti D. Pawar, MTech(2nd Year), C.I.I.T, Indore, damayantipawar09@gmail.com