Deny-by-Default Distributed Security Policy Enforcement in Mobile Ad Hoc Networks

Mansoor Alicherry[1], Angelos D. Keromytis[1], and Angelos Stavrou[2]

[1] Department of Computer Science, Columbia University
[2] Department of Computer Science, George Mason University

Abstract. Mobile Ad-hoc Networks (MANETs) are increasingly employed in tactical military and civil rapid-deployment networks, including emergency rescue operations and *ad hoc* disaster-relief networks. However, this flexibility of MANETs comes at a price, when compared to wired and base station-based wireless networks: MANETs are susceptible to both insider and outsider attacks. This is mainly because of the lack of a well-defined defense perimeter preventing the effective use of wired defenses including firewalls and intrusion detection systems.

We introduce a novel distributed security policy enforcement architecture that is designed specifically for MANETs. Our approach harnesses and extends the concept of *network capabilities* and is especially suited for mobile and heterogeneous communication environments. Our model imposes communication restrictions between MANET nodes by enforcing hop-by-hop policies in a distributed manner. We use a *deny-by-default* principle, allowing compromised nodes to access only authorized services. This significantly limits their ability disrupt or even interfere with end-to-end connectivity and nodes beyond their local communication radius. In this short paper, we only present the overall architecture of the system.

Keywords: MANETs, Capabilities, Distributed Firewall.

1 Introduction

Recent advances in low-power computing and communications have led to the proliferation of handheld and portable devices equipped with wireless connectivity. These mobile wireless devices appear to be ideal for situations where fixed infrastructure is too costly or dangerous to deploy, or has been rendered inoperable. However, because of radio power consumption, physical obstacles, and channel capacity, a mobile node may not be able to reach all other nodes within a single broadcast. Therefore, to achieve end-to-end connectivity, nodes have to form mobile *ad hoc* wireless networks (MANETs), which allow data to be routed through intermediate nodes. MANETs are fundamentally different from the Internet because all peers act as both sources and routers using the other participants to relay packets to their final destination. Due to their flexibility, MANETs are currently employed in both military and commercial applications.

Unfortunately, not all MANET nodes are equally capable, nor can all users be equally trusted. Worse yet, mobile nodes in tactical environments run the

Y. Chen et al. (Eds.): SecureComm 2009, LNICST 19, pp. 41–50, 2009.

danger of being captured or malfunction. Even a small number of misbehaving nodes can successfully render the entire MANET inoperable: malicious peers can abuse the network exhausting all network and power resources.

In traditional networks, malicious nodes and traffic are kept away from a set of nodes belonging to an organization or a group using *firewalls*. This is feasible because of the existence of a well defined network perimeter. All incoming and outgoing traffic needs to transit through these firewall nodes, which enforce the policies at the perimeter. Within the perimeter, smaller sub-groups can have more stringent policies by deploying their own firewalls. Unfortunately, the concept of a network perimeter does not exist in MANETs, and policies need to be enforced in a distributed manner while taking into consideration node mobility.

To address this, we propose an architecture that enforces trust relationships and traffic accountability between mobile nodes through a novel policy enforcement scheme designed specifically for MANETs. We extend the network capability framework [8,2] and we tailor it to the resource-constrained MANET environment. A capability is a token of authority that has associated rights. In our model, capabilities propagate both access control rules and traffic-shaping parameters that should govern a node's traffic. To that end, we define a protocol for communicating capabilities, which are treated as soft state, across the MANET.

Our architecture enables the enforcement of adaptive bandwidth constraints inside the network, denying by default unauthorized traffic. Nodes can only access the services and hosts they are authorized for by the capabilities given to them. Compromised or malicious nodes cannot exceed their authority and expose the whole network to an adversary. Upon detection, we can prevent a compromised node from further attacking the network simply by revoking its capabilities. Moreover, our architecture helps mitigate the impact of denial of service (DoS) attacks because excess or unauthorized packets are dropped closer to the attack source. Thus, we avoid unnecessary data processing and forwarding at the target node and the network itself.

Even though we focus on MANETs, our system can also be used in wired networks. However, MANETs provide our architecture both advantages and challenges. Specifically, the ratio of CPU cycles to available bandwidths (Hz/kbit) is normally higher in MANET nodes compared to their wired counterparts. This enables us to do more intelligent processing (and use cryptography) on most or all of the packets transiting through a MANET node. The number of traffic flows handled by a MANET node is also small due to the small network size. However, frequent route changes between a source and a destination node due to node mobility represents a difficult challenge in an distributed enforcement environment such as ours.

The rest of the paper is organized as follows. We begin by describing the threat model in Section 2. We then present the system architecture and a high-level overview of our scheme, including the security analysis, in Section 3. Related work is discussed in Section 4.

2 Threat Model

Our goal is to protect network resources and end-node services from denial of service attacks, and to enforce access control rules in the absence of a fixed topology. Thus, we want a node to be able to access only the services it is entitled to, and to limit the amount of traffic that can be sent to any such service. To preserve bandwidth and power, we need to filter any unauthorized traffic early on.

We assume MANET environments where an adversary may be an existing node that has been compromised (insider) or a malicious external node that might want to participate in the MANET. In addition, there may be multiple cooperating adversaries; and compromised nodes may not be detected as such immediately, or ever (depending on their actions).

The resources needed to access a service are allocated by the *group controller(s)* (GCs) of the MANET. Group controllers are nodes responsible for maintaining the group membership for a set of MANET nodes, and *a priori* authorize communications within the group. This means that GCs do not participate in the actual communications, nor do they need to be consulted by nodes in real time; in fact, if they distribute the appropriate policies ahead of time, they need not even be members of the MANET. In most cases, the GC may be reachable through a high-energy-consumption, high-latency, low-bandwidth long-range link (*e.g.,* a satellite connection); interactions in such an environment should be kept to a minimum, and only for exceptional circumstances (*e.g.,* for revoking access for compromised nodes).

Without compromising a GC, an external node can participate in a MANET only by stealing the authorization credentials that are bound to the identity of a legitimate node. Because we envision GCs as being primarily offline or, at best, intermittently reachable (with respect to the MANET), we are not addressing the issue of compromised controllers in this paper.

If a node is compromised, an adversary can only access the services and bandwidth that node is authorized to access. If other MANET nodes are adhering to our architecture, a compromised node does not have the ability to disrupt or interfere with end-to-end service connectivity and other nodes beyond its local radio communication radius. The nodes providing services will receive only the traffic that the compromised node is authorized to transmit, unless the adversary is in the local communication radius.

3 System Architecture

In our architecture, there is one or more pre-defined nodes that act as a *group controller* (GC). These nodes are trusted by all the group nodes. For simplicity and without loss of generality, we will assume that all the MANET nodes are part of a single group. A group controller has authority to assign resources to the nodes in MANET. These resources are expressed in terms of limits on the number of packets or on bandwidth rates that a MANET participant is permitted to transmit toward another node. The resource allocation by the GC to a node

is represented using a credential called *policy token* that all the nodes can verify. The policy tokens are typically provisioned ahead of time, and represent the projections of centralized policy, even though an on-demand allocation from the GC is possible. The GC may be offline after it distributes the policy tokens, and may be reachable sporadically at best after that (as external connectivity permits). The presence of the GC is not required, after the initial policy token distribution, for the normal working of the protocol.

When a node (initiator) requests a service from another MANET node (responder) using the policy token assigned to the initiator, the responder can provide a capability back to the initiator. This is called a *network capability*, and it is generated based on the resource policy assigned to the responder and its dynamic conditions (*e.g.,* level of utilization).

All the nodes in the path from an initiator to a responder (*i.e.,* nodes relaying the packets) are required to enforce and abide by the resource allocation encoded by the GC in the policy token and the responder in the network capability. The enforcement involves both accessibility and bandwidth allocation. A responder accepts packets (except for the first one) from an initiator only if the initiator has authorization to send, in the form of a valid network capability. An intermediate node will forward the packets from a node only if the packets have an associated policy token and network capability, and if they do not violate the conditions contained therein. Note that the possession of a network capability does not imply resource reservation; they are the maximum limit a node can use. Available resources are allocated by the intermediate nodes in a fair manner, in proportion to the allocations defined in the policy token and network capability. Intermediate nodes cache policy tokens and network capabilities in a *capability database*, treating them soft state.

Figure 1 gives an overview of the protocol exchanges when an initiator wants to communicate with a responder. The initiator has a policy token previously issued by the GC that authorizes the communication with the responder (step 1). The initiator sends a communication request (and, optionally, initial data), along with its policy token toward the responder (step 2). This packet also contains a *transaction id* that the initiator will use in subsequent packets to the same responder. The packet may also contain a network capability that the initiator generates; this can be used by the responder to communicate back to the initiator. Here, we assume that the initiator has a routing table entry for the responder. Otherwise the underlying routing protocol will be invoked to get the route. An intermediate node will forward the packet only after validating it (step 3). The validation involves cryptographic verification of the capability, and verification of the constraints (*e.g.,* bandwidth usage, service and destination address) specified in the policy token. If the validation is successful, the intermediate node also records the policy token in its capability database, along with other attributes of the packet, such as source and destination node address and the transaction id.

The responder, on receiving the packet verifies the policy token and creates a network capability for the initiator (step 4). The responder sends the response

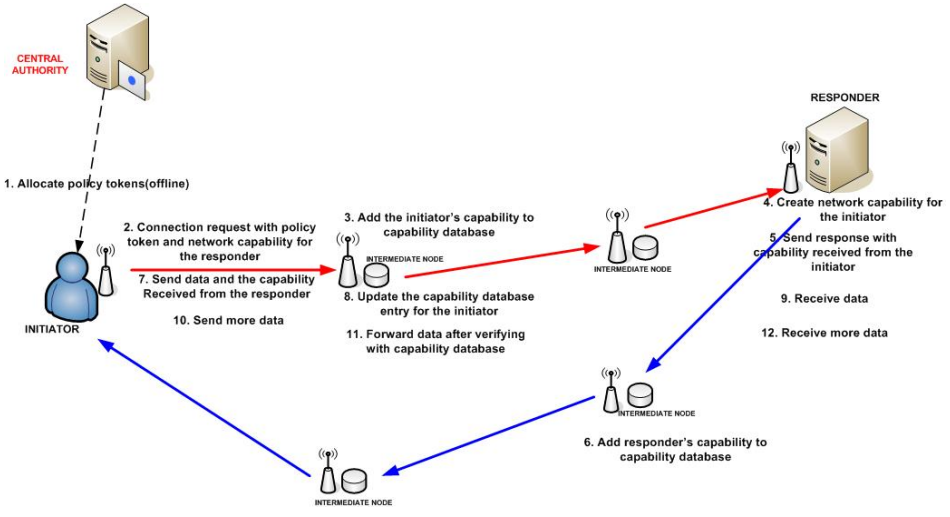

Fig. 1. System overview

to the request as well as the newly created network capability for the initiator (step 5). The responder also creates a transaction id for the communication, and includes it in the response. The responder also needs to include the network capability it received from the initiator in the first message, which authorizes it to communicate back; alternatively (or in addition), it may use a policy token issued by the GC to responder that is authorizing the communication with the initiator. Intermediate nodes, on receiving this packet from the responder, validate the packet and adds the responder's policy token and network capability to its capability database (step 6). In the diagram, the reverse path is shown to be different from the forward path; the paths can also be the same. The initiator will then have to include the responder-issued network capability in subsequent packets it transmits (step 7); intermediate nodes will add this credential to their capability database (steps 8, 9).

Any further data traffic between the initiator and the responder does not contain the policy token and network capability; instead, it contains only the transaction id that was included in the initial handshake (steps 10-12). The packets are signed by the sender, and can be verified by the intermediate nodes. If the cost of the cryptographic operations is too high (in terms of latency or power consumption), cryptographic validation may be done probabilistically. The intermediate nodes can validate the packets by looking at the policy token and network capability contained in the capability database corresponding to the transaction id in the packet. This process ensures that the packet does not exceed the resource limit allowed in the policy token and the network capability, and is authorized to reach the destination by both the GC and the destination itself. For this validation, the intermediate node also maintains the resource usage against each capability in its capability database. The only time the initiator or

responder need to re-send the capability is when the path between them changes due to node mobility, or when the network capability expires and is reissued by the peer.

We note that our solution can be used to protect multicast traffic and routing control packets. Furthermore, we can bound the probability of an adversary injecting traffic that remains undetected, when probabilistic cryptographic validation is performed. We omit the details due to lack of space.

3.1 Feasibility

We argue that the proposed solution is feasible for MANETs, even though the memory and processing power are lower in MANET nodes compared to routers in wired networks. Our scheme requires memory to store the information about the traffic sessions, and CPU cycles for the cryptographic operations. The feasibility comes from the fact that the bandwidth in MANETs is significantly lower than that of wired networks, while the nodes are relatively powerful (*e.g.,* normal laptops, or high-end cellphone devices). As a result, the available memory and processing power per packet is higher in MANETs than in wired networks. The processing power per packet for MANET nodes are increasing everyday with the advent of faster but less power-hungry processors for portable devices.

Furthermore, the per-packet cryptographic operations, which involve a public key signature verification, can be achieved with very small key sizes. This is because, unlike traditional uses of public keys, these keys are useful only for the short duration of the session. For longer sessions, new keys can be generated and old ones discarded.

3.2 Capability Definition

Each node has authority to send traffic to its peers at certain rates. This authority is encoded in the policy token and network capability. Both of these are represented by KeyNote-style credentials [3]. Each credential contains

1. Identity of the node (principal)
2. (Optional) Identity of the destination node; if left unspecified, it applies to all destinations
3. Type of service and amount of data the principal is allowed to send
4. An expiration time
5. Signature of the GC (for policy tokens) or peer (for network capabilities)

All nodes in the MANET know the public key of the GCs, so that they can verify policy tokens issued by them. Identities are expressed in term of the long-term public key of the node to which a credential is assigned. The destination node can be a host, subnet, or public key. Type of service refers to the transport protocol identifiers (*e.g.,* TCP ports) a credential authorizes.

Typically, the bandwidth available to a node on a network capability is higher than that of its policy token. Policy tokens are assigned by the GC, which has no knowledge of network load at the time the communication takes place. Hence,

the central authority will consider the worst case scenario while assigning the policy token and permit only enough communication to take place for a handshake to occur. It is up to the responder to provide a network capability with enough bandwidth allocation to enable the communication to proceed. Note, also, that it is in the interest of a node to issue short-lived network capabilities to its communicating peers, so that it can quickly respond to changing network dynamics or (more importantly) to peer misbehavior (*e.g.,* a flood-based DoS).

Policy tokens and network capabilities have the same syntactic representation. Following is an example:

```
serial: 130745
owner: unit01.nj.army.mil (public key)
destination: *.nj.army.mil
service: https
bandwidth: 50kbps
expiration: 2010-12-31 23:59:59
issuer: captain.nj.army.mil
signature: sig-rsa 23455656767543566678
```

The above represents a policy token assigned by node captain.nj.army.mil to unit01. The unit can use this policy token to send the traffic to any node in the domain *nj.army.mil*. The peak data rate using this credential cannot exceed 50kbps.

If unit01 wants to communicate with unit02, it will send a message to unit02 using this policy token. Unit02 will issue a network capability for unit01, if the communication needs more bandwidth than available in the policy token.

```
serial: 1567
owner: unit01.nj.army.mil (public key)
destination: unit02.nj.army.mil
bandwidth: 150kbps
expiration: 2007:10:21 13:05:35
issuer: unit02.nj.army.mil
comment: Policy allowing the receiver
    to issue this capability.
signature: sig-rsa 238769789789898
```

This capability is restricted to be used only by unit01 for communication with unit02. It specifies a higher bandwidth, but a shorter expiration date. The issuer of the capability is the same as the destination of the capability.

After receiving this capability, unit01 will use this capability for communication with unit02. The more general policy token can be used by unit01 for communicating with other nodes.

If the communication from unit01 to unit02 was short and required low bandwidth, unit01 could have used its policy token for the entire duration of the communication, without requesting for a network capability from unit02. This will be faster for short communication as there is no capability request/reply,

and unit02 does not have to issue any capabilities. If unit01 expects some messages from unit 2 that require more capabilities than the one that is available to unit02 in the form of its corresponding policy token, then unit01 could issue a network capability to unit02.

3.3 Security Analysis

We now discuss how our architecture relates to the threat model described in Section 2.

Since the capabilities are signed by a GC and are verifiable by all nodes, adversaries cannot generate their own valid capabilities. Adversaries can create valid capabilities only if the GC is compromised. Since the individual packets are signed, an adversary cannot use a transaction id that does not belong to it to transmit packets.

A compromised or malicious node that does not enforce the capability protocol can only have impact within its communication radius. Packets generated without the capability or with a snooped transaction id by a malicious node will be dropped by the neighboring nodes due to invalid signatures. A compromised node can only access the services it is authorized to. Packets of nodes trying to use more bandwidth than is allocated to them will be rejected. A malicious node frequently doing this can be detected and isolated.

A receiver can protect against DoS attacks by controlling the issuance of network capabilities to its peers. A malicious node can use its policy tokens or network capabilities to send duplicate packets in multiple disjoint paths; we do not currently protect against this attack, which allows a node to transmit more traffic that it is authorized to. We note, however, that local nodes in the radio perimeter of the misbehaving node can detect this scenario. Since the network capability can be created only based on the policy allowed by the GC, it is not possible for two compromised nodes to collaborate and create arbitrarily large network capabilities.

4 Related Work

Security for mobile *ad hoc* network is an active area of research. Most of the prior work on MANET security focused on solving specific problems or retrofitting security into an existing IP-based network architecture; we are trying to introduce a new architecture where security is built into the network. Surveys of research in MANETs can be found elsewhere [11,13,9].

The concept of capabilities was used in operating system for securing resources [10]. There was work on allowing controlled exposure of resources at the network layer using the concept of "visas" for packets [4], which is similar to network capabilities. More recently, network capabilities were proposed to prevent DoS in wired networks [2]. We extend the concept to MANET and use it for both access control rules and traffic shaping parameters. In the original approach, the capabilities were assigned only by the receivers, and there is no limit on the

amount of capability that a receiver can assign. Though it achieves the goal of preventing the DoS attack at the receiver, it does not prevent two nodes from taking up all the available network resources. Their solution also assumes that the links in the path between a sender and receiver cannot be snooped, and the path is fixed. These assumptions are valid for the wire line system that their solution is designed for, but does not work for MANETs. Previous work on distributed firewalls [5] focused on wired fixed-network environments, and attempts to protect only the end hosts using a host-based solution. Our solution is for a mobile network, using a combination of network and host-based solutions that attempt to protect both the network and end-host resources.

Signing and verification of packets between a sender and a receiver were commercially available in early 1990s. Novell's Netware 3.11 and 4.x supported *NCP Packet Signature Option*, where a unique signature was appended to each packet sent between the client and the server [7]. The keys for the signatures were negotiated at login time. Intermediate nodes were not involved in packet verification.

Mitigating the denial of service attacks by including a message authentication code and the certificate of the sender for each packet has been previously proposed [12]. That work does not study the high overhead associated with sending a large signature or a large certificate on each packet. The authors use game theory to study the problem of dealing with selfish nodes that do not verify the packet signatures, using incentives and punishments. This mechanism or any other reputation based mechanism [6] can also be used in our scheme to deal with selfish nodes.

HEAP [1] mitigates various MANET attacks from outsider nodes by doing a hop-by-hop packet authentication using HMAC. MACs (end-to-end or hop-by-hop) cannot deal with insider attacks. They also cannot provide access control unless different MAC keys are used for different policies. Even with different keys, MACs allow rogue nodes to "hide" since MACs are repudiable as all the intermediate nodes in the path between a sender and a receiver need to know the key. Only asymmetric key mechanisms can allow validation by all the intermediate nodes that the packets indeed sent by the source node of the packet.

5 Conclusions and Future Work

We presented a novel architecture for enforcing security policies in MANETs. Our scheme, based on the concept of network capabilities and following a deny-by-default paradigm, can protect both end-host resources and network bandwidth from denial of service attacks, as well as limit the exposure of the MANET to compromised and malicious nodes. We discussed the details of the architecture and protocol used for propagating policy tokens and receivers, and discussed the various scenarios of use. For our future work, we plan to study the impact of our scheme on throughput and latency for different topologies and classes of traffic. In addition, we intend to quantify the performance of multicast traffic on mobility scenarios, and to implement and deploy on MANET testbeds with real traffic.

Acknowledgements

This work was supported in part by the National Science Foundation through Grant CNS-07-14277. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF or the US Goverment.

Mansoor Alicherry was supported by Alcatel-Lucent, Murray Hill, New Jersey.

References

1. Akbania, R., Korkmaz, T., Raju, G.: HEAP: A packet authentication scheme for mobile ad hoc networks. In: Communications and Networking Simulation Symposium (2007)
2. Anderson, T., Roscoe, T., Wetherall, D.: Preventing internet denial-of-service with capabilities. In: Proc. of Hotnets-II (2003)
3. Blaze, M., Ioannidis, J., Keromytis, A.: Trust management for ipsec. In: Symposium on Network and Distributed Systems Security, SNDSS (2001)
4. Estrin, D., Mogul, J.C., Tsudik, G.: Visa protocls for controlling interorganizational datagram flow. IEEE Journal on Selected Areas in Communications (May 1989)
5. Ioannidis, S., Keromytis, A.D., Bellovin, S.M., Smith, J.M.: Implementing a distributed firewall, pp. 190–199 (2000)
6. Jaramillo, J., Srikant, R.: Darwin: Distributed and adaptive reputation mechanism for wireless ad-hoc networks. In: MOBICOM (2007)
7. Lee, R.: Netware 4.x performance tuning and optimization: Part 3 (October 1993), `http://support.novell.com/techcenter/articles/ana19931001.html`
8. Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Hu, Y.-C.: Portcullis: protecting connection setup from denial-of-capability attacks. SIGCOMM Comput. Commun. Rev. 37(4), 289–300 (2007)
9. Shi, E., Perrig, A.: Designing secure sensor networks. IEEE Wireless Communications (2004)
10. Wobber, E., Abadi, M., Burrows, M., Lampson, B.: Authentication in the taos operating system. ACM Transactions on Computer Systems 12 (February 1994)
11. Wu, B., Chen, J., Wu, J., Cardei, M.: A survey on attacks and countermeasures in manets. In: Wireless/Mobile Network Security, ch. 12. Springer, Heidelberg (2006)
12. Wu, X., Yau, D.K.Y.: Mitigating Denial-of-Service Attacks in MANET by Distributed Packet Filtering: A Game-theoretic Approach. In: ASIACCS (March 2007)
13. Yang, H., Luo, H., Ye, F., Lu, S., Zhang, L.: Security in mobile ad hoc networks: Challenges and solutions. IEEE Wireless Communications (2004)