International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com, Special Issue 42 (AMBALIKA) (March 2017), PP. 16-20

CAUSE & EFFECT OF RUSHING ATTACK IN WIRELESS NETWORK

[1] Er. UPENDRA CHAUHAN
[2]Er. SUNIL KUMAR VISHWAKARMA
[1,2] Assistant Professor
Department of Computer science & Information Technology,
Ambalika institute of Management & Technology, Lko
[1]cse.upendra@gmail.com, [2]sunilvishwakarma83@gmail.com

***Abstract*— The rushing attack, a new attack that results in denial-of-service when used against all previous on-demand ad hoc network routing protocols, For example, DSR, AODV, and secure protocols based on them, such as Ariadne, ARAN, and SAODV, are unable to discover routes longer than two hops when subject to this attack. This attack is also particularly damaging because it can be performed by a relatively weak attacker. We analyze why previous protocols fail under this attack. We then develop Rushing Attack Correction (RAC), a generic defense against the rushing attack for on-demand protocols. RAC incurs no cost unless the underlying protocol fails to find a working route, and it provides provable security properties even against the strongest rushing attackers.**

***Index Terms*—** Cause & Effect, rushing attack.

I. INTRODUCTION

In this paper, we present a new attack, the rushing attack, which results in denial-of-service when used against all previously published on-demand ad hoc network routing protocols. Specifically, the rushing attack prevents previously published secure on-demand routing protocols to find routes longer than two-hops (one intermediate node between the initiator and target). Because on- demand protocols generally have lower overhead and faster reaction time than other types of routing based on periodic (proactive) mechanisms, on-demand protocols are better suited for most applications. To defend this important class of protocols against the rushing attack, we develop a generic secure Route Discovery component, called Rushing Attack Correction (RAC), that can be applied to any existing on-demand routing protocol to allow that protocol to resist the rushing attack. Our main contributions in this paper are the presentation of the rushing attack, the development and analysis of our new secure Route Discovery component that demonstrates that it is possible to secure against the rushing attack, and a general design that uses this component to secure any on-demand Route Discovery mechanism against the rushing attack.

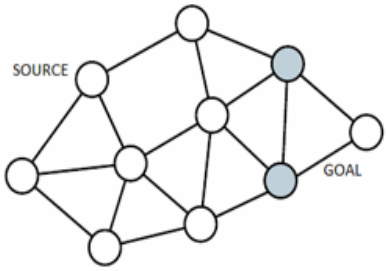

Figure: 1.1 Example of Rushing Attack on network

II. THE RUSHING ATTACK AGAINST AD HOC NETWORK ROUTING PROTOCOLS

The rushing attack that acts as an effective denial- of-service attack against all currently proposed on- demand ad hoc network routing protocols, including protocols that were designed to be secure. In an on-demand protocol, a node needing a route to a destination floods the network with ROUTE REQUEST packets in an attempt to find a route to the destination. To limit the overhead of this flood, each node typically forwards only one ROUTE REQUEST originating from any Route Discovery. In particular, existing on-demand routing protocols, such as AODV, DSR, LAR, Ariadne, SAODV, ARAN, AODV, SUCV and SRP only forward the REQUEST that arrives first from each Route Discovery. In the rushing attack, the attacker exploits this property of the operation of Route Discovery. The initiator node initiates a Route Discovery for the target node. If the ROUTE REQUESTs for this Discovery forwarded by the attacker are the first to reach each neighbor of the target, then any route discovered by this Route Discovery will include a hop through the attacker. That is, when a neighbor of the target receives the rushed REQUEST from the attacker, it forwards that REQUEST, and will not forward any further REQUESTs from this Route Discovery. When non-attacking REQUESTs arrive later at these nodes, they will discard those legitimate REQUESTs. As a result, the initiator will be unable to discover any usable routes (i.e., routes that do not include the attacker) containing at least two hops (three nodes).

A rushing attacker need not have access to vast resources. On demand routing protocols delay ROUTE REQUEST

forwarding in two ways. First, Medium Access Control (MAC) protocols generally impose delays between when the packet is handed to the network interface for transmission and when the packet is actually transmitted. In a MAC using time division, for example, a node must wait until its time slot to transmit, whereas in a MAC using carrier-sense multiple access, a node generally performs some type of back-off to avoid collisions; protocols like IEEE 802.11 also impose an inter-frame spacing time before transmission actually begins.
Second, even if the MAC layer does not specify a delay, on-demand protocols generally specify a delay between receiving a REQUEST and forwarding it, in order to avoid collisions of the REQUEST packets. In particular, because REQUEST packets are broadcast and collision detection for broadcast packets is difficult, routing protocols often impose a randomized delay in REQUEST forwarding. An attacker ignoring delays at either the MAC or routing layers will generally be preferred to similarly situated non- attacking nodes. One way to thwart an attacker that rushes in this way is to remove these delays at both the MAC and routing layers, but this approach does not work against all types of rushing attackers and is not general. For example, in a dense network using a CSMA MAC layer, if a node A initiates a Route Discovery, and B is two hops away from A, and C and D are neighbors of both A and B, then then B will likely not receive the ROUTE REQUEST due to a collision between REQUESTs forwarded by C and D. In a dense network, such collisions may often prevent the discovery of any nontrivial routes (routes longer than a direct link), which is even more severe than the rushing attack, which prevents the discovery of routes longer than two hops. Another way that a relatively weak attacker can obtain an advantage in forwarding speed is to keep the network interface transmission queues of nearby nodes full. For example, if each node processes the packets it receives in order, and an inefficient REQUEST authentication mechanism is used, the attacker can keep other nodes busy authenticating REQUESTs containing bogus authentication, thus slowing their ability to forward legitimate REQUESTs. Protocols employing public key techniques are particularly susceptible to these attacks, since they require substantial computation to validate each received REQUEST. A relatively weak attacker can also achieve faster transit of its REQUEST packets by transmitting them at a higher wireless transmission power level, thus reducing the number of nodes that must forward that REQUEST to arrive at the target. Since packet transit time at each hop is dominated by the processing time at the forwarding node, reducing the path to the target by just one hop is likely to provide a significant latency advantage, thus strengthening the attacker's position.

III. SECURE ROUTING REQUIREMENTS AND PROTOCOL

a set of generic mechanisms that together defend against the rushing attack: secure Neighbor Detection, secure route delegation, and randomized ROUTE REQUEST forwarding. We also describe a technique to secure any protocol using an on-demand Route Discovery protocol.
In previous on-demand protocols, node B considers node A to be a neighbor when B receives a broadcast message from A. Secure Neighbor Detection, which replaces standard Neighbor Detection, allows each neighbor to verify that the other is within a given maximum transmission range. Once a node A forwarding a ROUTE REQUEST determines that node B is a neighbor (that is, is within the allowable range), it signs a Route Delegation message, allowing node B to forward the ROUTE REQUEST. When node B determines that node A is within the allowable range, it signs an Accept Delegation message. Randomized selection of the ROUTE REQUEST message to forward, which replaces traditional duplicate suppression in on-demand route discovery, ensures that paths that forward REQUESTs with low latency are only slightly more likely to be selected than other paths.
Figure 3 shows the basic design of our complete rushing attack correction mechanism.

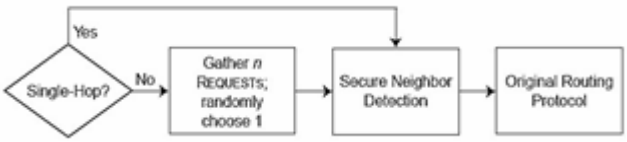

Figure: 3 RAC Mechanisms

A. NOTATION

- A or B denote communicating nodes
- $A : \eta \xleftarrow{R} \{0,1\}^{l}$ denotes that node A randomly selects an l-bit long nonce η
- $A \rightarrow B : \langle M, H(A \,||\, \eta) \rangle$ means that node A sends B the message M and the hash of A's identifier concatenated with the nonce η.
- $A \rightarrow * : \langle M, \Sigma_M \rangle$ means that node A broadcasts message M with its signature Σ_M.

B. SECURE NEIGHBOR DETECTION

a secure Neighbor Detection protocol that allows both the sender and the receiver of a ROUTE REQUEST to verify that the other party is within the normal direct wireless communication range. The functionality of Neighbor Detection, in which two nodes detect a bidirectional link between themselves, is present in some form in almost every routing protocol.

Requirements for Secure Neighbor Detection Two nodes detect each other as neighbors only if they can communicate and they are within some maximum transmission range.

The secure Neighbor Detection protocol thus prevents an attacker from: (1) introducing two nodes that are not within the maximum transmission range as neighbors; and (2) claiming

that it is a neighbor of another node without being able to hear packets directly from that node.

1) Secure Neighbor Detection Protocol

We present a secure Neighbor Detection protocol that allows both the initiator and the responder to check that the other is within a maximum communication range. Finally we rate-limit new neighbor solicitations to prevent an attacker from flooding its neighbors Figure Below shows the full protocol.

2) Integration with an On-Demand Protocol

In an on-demand protocol, neighbor verification is performed during each Route Discovery. As a result, we can defend against New Neighbor Solicitation floods, by relying on the underlying protocol to defend against ROUTE REQUEST floods; a node responds to any New Neighbor

Neighbor Detection between initiator S and responder R

S:	$\eta 1 \leftarrow R \leftarrow \{0,1\}^l$
	M1=<NEIGHBOUR SOLICITATION, S, η1>
	$\Sigma_M = Sign\ (H(M_1))$
$S \rightarrow *:$	$<M_1, \Sigma_{M\ 1}>$
R:	$\eta 2 \leftarrow R \leftarrow \{0,1\}^l$
	M2=<NEIGHBOUR REPLY, S, R, η1, η2)>
	$\Sigma_{M\ 2} = Sign\ (H(M_2))$
$R \rightarrow S:$	$<M_2, \Sigma_{M\ 2}>$
S:	***M3=<NEIGHBOUR VERIFICATION, S, R, η1, η2>***
	$\Sigma_{M\ 3} = Sign\ (H(M_3))$
$S \rightarrow R:$	$< M_3, \Sigma_{M\ 3}>$

When a node *A* forwards a REQUEST, it includes in that REQUEST a broadcast Neighbor Solicitation. Each node *B* forwarding that REQUEST returns a Neighbor Reply, and piggybacks on the Neighbor Reply a unicast Neighbor Solicitation for *A*. If *A* decides that *B* is a neighbor based on the wormhole prevention mechanism used, *A* returns a signed Neighbor Verification that verifies the link from *A* to *B*. *A* also includes in packet a Neighbor Reply to the unicast Neighbor Solicitation sent by *B*. If *B* decides that *A* is a neighbor based on the wormhole prevention mechanism used, *B* forwards the REQUEST, including the Neighbor Verification for the $A \rightarrow B$ link signed by *A*, and also including a Neighbor Verification for the $B \rightarrow A$ link signed by itself. *B* need not return a Neighbor Verification, since *A* is likely to hear the forwarded REQUEST, which includes the $B \rightarrow A$ Neighbor Verification.

Figure 4 shows how *B* forwards a REQUEST from *A*.

C. SECURE ROUTE DELEGATION

We use this mechanism to enable the nodes to verify that all the secure neighbor detection protocols were executed and that both neighbors believe that they are within transmission range. We describe the protocol based on an example. Consider two neighboring nodes A and B, where A received the current ROUTE REQUEST originating from node S destined for node R with the sequence number id. Node A engages in the secure neighboring detection protocol and finds after the second message that B is indeed within range, so it delegates the ROUTE REQUEST to B as follows:

MA	**=<ROUTE DELEGATION, *A*, *B*, *S*, *R*, *id*>**
Σ_{MA}	$=Sign(H(M_A))$
$A \rightarrow B:$	$<\Sigma_{MA}>$

Node A does not need to send the message to B, as B can reconstruct all the fields of the message and verify the signature. The ROUTE DELEGATION message can be bundled together with the last message of the secure Neighbor Detection protocol. If B believes that A is indeed a neighbor within range, B will accept the ROUTE DELEGATION, continue the protocol, and sign another ROUTE DELEGATION with the next neighbor.

D. RANDOMIZED MESSAGE FORWARDING

The secure Neighbor Detection and secure Route Delegation techniques are not sufficient to thwart the rushing attack, since an adversary can still get an advantage by forwarding ROUTE REQUESTs very rapidly. We use a random selection technique to minimize the chance that a rushing adversary can dominate all returned routes. In traditional ROUTE REQUEST forwarding, the receiving node immediately forwards the REQUEST and suppresses all subsequent REQUESTs. In our modified flooding, a node first collects a number of REQUESTs, and selects a REQUEST at random to forward. There are thus two parameters to our randomized forwarding technique: first, the number of REQUEST packets to be collected and second, the algorithm by which timeouts are chosen.

E. SECURE ROUTE DISCOVERY

In this section, we describe our secure route discovery protocol. We use three techniques in concert to prevent the rushing attack: our secure Neighbor Discovery protocol, our secure Route Delegation and delegation acceptance protocol, and randomized selection of which ROUTE REQUEST will be forwarded. The intuition behind Secure Route Discovery is to make the forwarding of REQUEST packets less predictable by buffering the first n REQUESTs received, then randomly choosing one of those REQUESTs. However, we need to prevent an attacker from filling too many of these n REQUESTs, since otherwise the attacker could simply rush n copies of a REQUEST, rather than a single REQUEST, and our scheme would once again be vulnerable to the rushing attack.

F. INTEGRATING SECURE ROUTE DISCOVERY WITH DSR

To integrate rushing prevention with DSR or other secure protocols based on DSR, we limit Route Discovery frequency as in Ariadne. Each time a node forwards a ROUTE REQUEST, it first performs a Secure Neighbor Detection exchange with the previous hop. When it forwards the REQUEST, it includes in the REQUEST a bidirectional Neighbor Verification for the previous hop. As in DSR, the target of a Route Discovery returns a ROUTE REPLY for each distinct ROUTE REQUEST it receives. Each such

ROUTE REPLY is sent with a source route selected by reversing the route in the ROUTE REQUEST. This route is likely to work if there are no attackers on the route, since Neighbor Detection only finds bidirectional neighbors.

G. SECURITY ANALYSES

This section discusses the security properties achieved with RAC when n distinct routes (both legitimate and attacking) exist between the originator and each other node in the network.

Since routes are required to end in different nodes, an attacker with access to the keys of m compromised nodes can generate at most m distinct, maliciously injected ROUTE REQUESTs for the purpose of denial-of-service. To analyze the probability of a node subverting a Route Discovery, we assume that the attacker rushes m distinct REQUESTs to each node in the network. As a result, each node needs only n-m additional distinct REQUESTs. We also suppose that the network

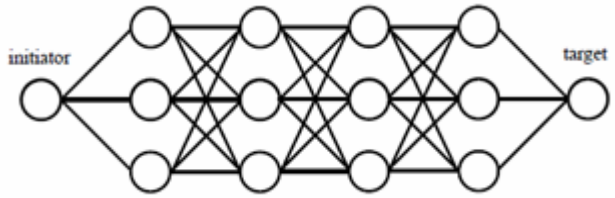

Figure 3.1 Example network topology used in RAC Security analysis

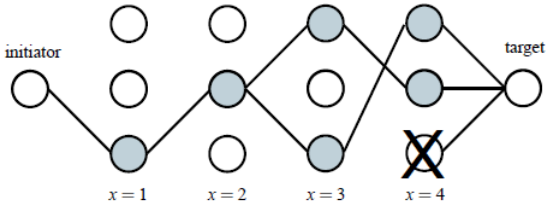

Figure 3.2 An example of a successful Route Discovery. Each gray node chose a valid REQUEST and belonged to a route for which a REPLY was sent. Each line represents a hop in a path chosen by a legitimate REQUEST; the network topology is shown in Figure 4.1

CONCLUSION

n this paper, we have described the rushing attack, a novel and powerful attack against on- demand ad hoc network routing protocols. This attack allows an attacker to mount a denial-of- service attack against all previously proposed secure on-demand ad hoc network routing protocols.

$$S_{x,y} = \sum_{i=1}^{y}\left[\binom{y}{i}\left(\frac{m}{n}\right)^{y-i}\left(\frac{n-m}{n}\right)^{i}\sum_{j=1}^{\min(i,n-m)} S_{x-1,j}\,p_{n-m-j}(n-m,i)\right]$$

$$= \sum_{i=1}^{y}\left[\binom{y}{i}\left(\frac{m}{n}\right)^{y-i}\left(\frac{n-m}{n}\right)^{i}\sum_{j=1}^{\min(i,n-m)}\left\{S_{x-1,j}\binom{n-m}{n-m-j}\sum_{k=0}^{j}(-1)^{k}\binom{j}{k}\left(\frac{j-k}{n-m}\right)^{i}\right\}\right]$$

Figure 5: The probability of a successful Route Discovery in a network using RAC

We have also presented RAP (Rushing Attack Prevention), a new protocol that thwarts the rushing attack. We found that the widely used duplicate suppression technique makes the rushing attack possible, and we designed a new Route Discovery protocol called RAP that replaces the standard mechanism and thwarts the rushing attack. Our approach is generic, so any protocol that relies on duplicate suppression in Route Discovery can use our results to fend off rushing attacks. More importantly, we demonstrated that there are mechanisms that can defend against the rushing attack, even though all previous attempts at secure on-demand ad hoc network routing protocols have been vulnerable.

REFERENCES

[1] Bradley R. Smith and J.J. Garcia-Luna- Aceves. Securing the Border Gateway Routing Protocol. In Global Internet'96, London, UK, November 1996.

[2] Bradley R. Smith, Shree Murthy, and J.J. Garcia-Luna-Aceves. Securing Distance Vector Routing Protocols. In Symposium on Network and Distributed Systems Security (NDSS'97), February 1997.

[3] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In Security Protocols, 7th International Workshop, edited by B. Christianson, B. Crispo, and M. Roe. Springer Verlag Berlin Heidelberg, 1999.

[4] Richard von Mises.U¨ ber Aufteilungs- und Besetzungswahrscheinlichkeiten. Revue de la Facult´e des Sciences de l'Universit´e d'Istanbul, 4:145—163, 1939.

[5] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-Aware Ad-Hoc Routing for Wireless Networks. Technical Report UIUCDCS- R-2001-2241, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001.

[6] Manel Guerrero Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In Proceedings of the ACM Workshop on Wireless Security (WiSe 2002), September 2002.

[7] Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. IEEE Network Magazine, 13(6), November/December 1999.

[8] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In Seventh Annual ACM International Conference on Mobile Computing and Networks (MobiCom 2001), Rome, Italy, July 2001.