FLOODING ATTACK ON THE BINDING CACHE IN MOBILE IPv6

Christian Veigner[1], Chunming Rong
University of Stavanger, N-4036, Norway

Keywords: Binding Cache flooding attack, Mobile IPv6, Return Routability, ROM.

Abstract: In the next generation Internet protocol (IPv6), mobility is supported by means of Mobile IPv6 (MIPv6). As a default part of the MIPv6 protocol, route optimization is used to route packets directly to a mobile node's currently used address at the mobile node's visited subnet. Return Routability is the protocol suggested by the IETF for managing this task. Route optimization is often carried out during handovers, where a mobile node changes network attachment from one subnet to another. To offer seamless handovers to the user it is important that route optimizations are carried out quickly. In this paper we will present an attack that was discovered during design of a new and more seamless protocol than the Return Routability. Our improved route optimization protocol for Mobile IPv6 suffers this attack; therefore we wanted to investigate if a similar attack was feasible on the Return Routability protocol. In this paper, we show that our new route optimization protocol offers no less security than the already standardized Return Routability protocol in this field.

1 INTRODUCTION

Route optimization is introduced to the *Mobile IPv6* (MIPv6) protocol (Johnson, 2004). However, it is important that the new feature doesn't result in new vulnerabilities to the IPv6 protocol. If not properly designed, it is believed that certain attacks on this optimization protocol could cause serious problems to the stability of the entire Internet. Hence, it is most important to investigate different attacks and their countermeasures.

Authentication of *mobile nodes* (MNs) is one of the most important features of such a mobility protocol. Initially, strong authentication was thought to be the only solution, and IPsec was at some point of time believed to be the best fit for this purpose. Due to the fact that IPsec, in addition to other protocols that relies on additional infrastructure, is not very scalable, the strong authentication demand evolved into a weaker authentication demand. The lack of scalability when using IPsec, stem from the key exchange necessity of each pair of communicating nodes. The protocol finally suggested by the IETF was the *Return Routability* (RR) (Johnson, 2004).

[1]This work was supported by UiS 95310, Rogaland University Fund.

As an example, RR decreases an attacker's range of launching a redirecting attack (Deng, 2002) from the entire Internet, to the necessity of being on the route between MN's *home agent* (HA) and one of MN's *corresponding nodes* (CNs). The HA is a node at MN's home subnet that cooperates with MN when MN is visiting a foreign subnet. Any other node communicating with MN is referred to as a CN. The redirecting attack is possible due to weak authentication. However, this is a huge improvement, reducing an attacker's range from the entire Internet to the HA-CN route, without the need of any additional infrastructure.

Focusing on the main drawback of the RR protocol, the possibility of experiencing lack of seamless handovers, we designed *a new route optimization protocol for Mobile IPv6 (ROM)* (Veigner, 2004). This protocol intends to decrease the latency of route optimization when actually needed, that is, when the MN suddenly changes subnet.

We investigate (Veigner, 2004) to which extent the ROM protocol suffers from redirecting attacks (Deng, 2002), bombing attacks (Aura, 2002), amplification attacks (Aura, 2002) and flooding attacks. Flooding attacks on route optimization protocols in general are briefly described in (Nikander, 2005).

Veigner C. and Rong C. (2005).
FLOODING ATTACK ON THE BINDING CACHE IN MOBILE IPv6.
In *Proceedings of the Second International Conference on e-Business and Telecommunication Networks*, pages 36-43
DOI: 10.5220/0001414400360043

During analyses of the ROM protocol design, we discovered that flooding attacks on a corresponding node's (CN's) *binding cache* (BC) easily might be carried out. A BC is a cache allocated at CN's for storing bindings between home and foreign addresses of mobile nodes. A *flooding attack* aims to fill such BCs with spurious entries. A CN may thereby be unable to perform route optimization with new MNs.

In this paper we will further describe flooding attacks in detail, and also show to which extent such attacks are feasible on the Return Routability (RR) protocol as well as on our ROM protocol.

Even though the RR protocol does not store any state at a CN before the initiating MN's authenticity is verified, we will show that flooding attacks on a CN's BC are possible.

The rest of this paper is organized as follows. In Section 2, an introduction to route optimization and the binding cache (BC) is given. Section 3 introduces our ROM protocol design and exemplifies the BC flooding attack. Section 4 focuses on the RR protocol and elaborates the possibilities of similar BC flooding attacks on the RR protocol. Finally our paper is concluded in Section 5.

2 ROUTE OPTIMIZATION AND THE BINDING CACHE

The key advantage of *route optimization* is a corresponding node's (CN's) ability to continue its session with a MN over an optimal route, even when the MN changes its point of attachment to the Internet. Now the MN's home agent (HA) does not have to reroute all of MN's incoming packets to MN's dynamically changing location. Due to route optimization, latency of data transmissions and bandwidth misuse may be substantially reduced.

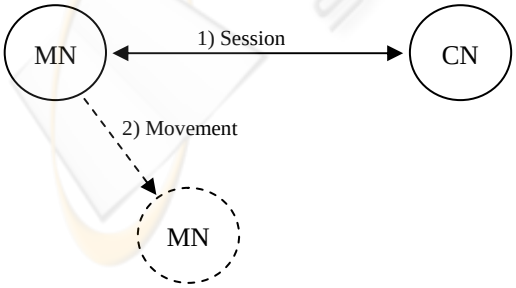

Figure 1: Movement of a mobile node.

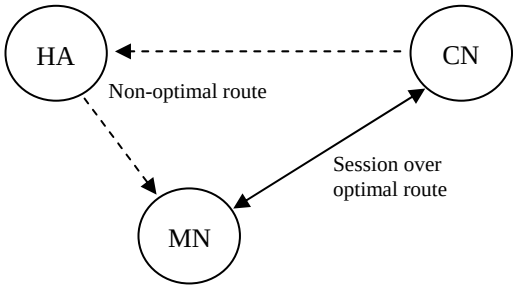

Figure 2: Route optimization.

A MN having an ongoing session with a CN is shown in figure 1. If the MN moves to another subnet (figure 1), the packets should be routed directly from CN to MN as shown in figure 2 (route optimization). The alternative suboptimal solution is seen as dotted lines in figure 2. The other way around however, sending packets directly from MN to CN (also shown in figure 2), is a problem solved long ago (Johnson, 2004), and will hence not be discussed in this paper.

By means of route optimization, only the initial packets from a CN may be routed through the MN's HA. This may occur if the CN has no entry of the receiving MN in its binding cache (BC). The CN thereby assumes that MN is situated at its home subnet. Whenever a packet forwarded by the HA arrives at the MN, MN initiates route optimization, informing the CN of its current location. The remaining packets from CN may from now on be routed directly to MN.

We will now give a brief introduction to the binding cache (BC) located at CNs. Mobile and fixed nodes are not differentiated in IPv6; thereby a packet-sending node always has to check its BC for an entry of the receiving node before a packet is transmitted. If an entry exists, the transmitting node must route its packets directly to the MN's care-of address (CoA).

Generally, a BC contains *home addresses* (HoAs) and *care-of addresses* (CoAs) of mobile nodes (MNs). This is shown in figure 3. A HoA is the address of a MN at its home subnet and a CoA is the address currently associated with the MN at its visited subnet. This information is contained in a CN's BC for each of the MNs that the CN has been in contact with recently.

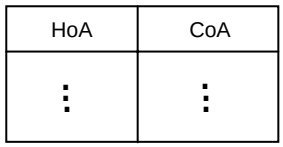

Figure 3: Binding cache.

Additionally, the BC will often maintain remaining lifetime of these bindings, and maybe the highest received sequence number associated with each binding. The sequence numbers may be used for replay attack prevention. Both CNs and HAs must be able to allocate memory for a BC.

We will in this paper focus on the BC at CNs, and elaborate the possibility of launching a BC flooding attack, filling the BC with non-real bindings of non-existing MNs. Since every node in MIPv6 may become a CN to a MN, and the MIPv6 protocol is supposed to be a default part of the IPv6 protocol, every IPv6 node must able to allocate memory resources for a BC.

Even a small handheld unit may become a CN. Such units are normally equipped with quite limited memory resources, and may easily become targets of BC flooding attacks.

3 THE ROM PROTOCOL

In this section an overview of our ROM protocol is given. The protocol is described in more depth in (Veigner, 2004). The ROM protocol is supposed to be an alternative to, and a more seamless protocol than the IETF Return Routability (RR) protocol (Johnson, 2004). Hopefully, ROM offers security characteristics similar to the RR protocol.

A MN uses the ROM protocol to assign a unique hash value to its currently used CN. The hash value is sent via the HA. Simultaneously the home subnet of MN is authenticated by the CN by means of a three-way handshake. When moving into a new subnet, MN now only has to send a *binding update* (BU) message directly to the CN. The CN considers the BU message authentic due to MN's knowledge of the nonce value. The nonce value included in the BU message was previously used when generating CN's unique hash value. Routing packets over the optimal route may now begin.

The main part of the ROM protocol messages is shown in figure 4. These messages are sent in advance of MN's movement to a new subnet. The messages shown in figure 5 are sent as the final part of the handover procedure when MN arrives at its new subnet.

We will now introduce the messages of our ROM protocol. We'll start with the messages of figure 4. For more in depth explanation, see (Veigner, 2004).

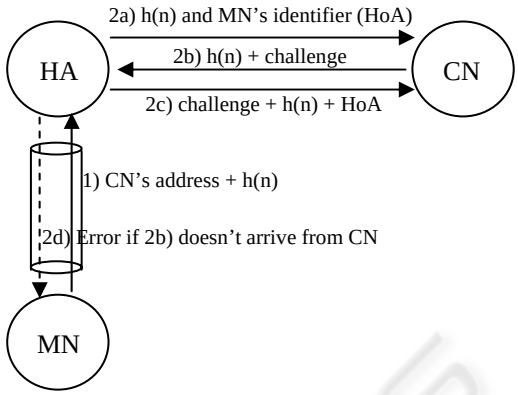

Figure 4: The ROM protocol.

Message 1: This message is sent by the MN to its HA. It contains the address of a CN and a unique hash value (h(n)). This message might of course contain a *list* of CNs and unique hash values for increased efficiency.

Message 2a: The received hash value is sent to the CN's address, along with MN's identifier (HoA). The source address of the 2a message is the address of the HA.

Message 2b: CN returns the hash value and includes a challenge for the HA.

Message 2c: HA returns the challenge, and once again the hash value is sent along with MN's HoA address. CN may thereby remain stateless until the MN's home subnet is authenticated by the 2a - 2c procedure.

Message 2d: If the HA doesn't receive a 2b message in reply of a 2a message, HA notifies MN of CN's absence. It is now in vain to proceed with the route optimization protocol with this CN.

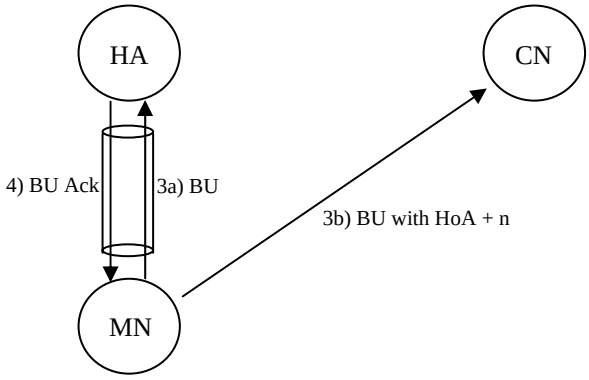

Figure 5: BU messages sent from MN's new location.

The messages of figure 5 are as follows.

Message 3a: An ordinary BU message is sent to MN's HA.

Message 3b: A BU message is also sent to the CN. This message contains MN's identifier (HoA) and the nonce value previously used when CN's unique hash value was generated. The source address of both the 3a and 3b messages is the MN's CoA address.

Message 4: As in the RR protocol, HA must always return a BUAck message.

Whenever a CN receives a BU message from a MN, the hash value used for authenticating the MN is deleted from its cache. A new hash value must now be assigned to the CN, otherwise CN will be unable to authenticate MN's next BU message, if one is ever to arrive.

We will in the following introduce a BC flooding attack on the ROM protocol.

3.1 BC flooding attack on the ROM protocol

A BC flooding attack aims to flood a CN's BC with spurious bindings of non-existing nodes. A CN, which may be any node in an IPv6 network, must be able to allocate memory resources for this BC, mapping home addresses (HoAs) to care-of addresses (CoAs).

We will not go into all the details of our ROM protocol design in this paper, but rather focus on the possible binding cache (BC) flooding attack. Later on, in Section 4.1 and Section 4.2, we will show to which extent a similar attack may be launched on the RR protocol.

Due to the three-way handshake of the ROM protocol, an Eve may attack a CN from anywhere in the entire Internet.

As shown in figure 6, we may consider an Eve transmitting a 2a message to its victim CN. On reception of the 2b message from the attacked node, Eve replies with a 2c message. By repeatedly doing this, Eve may be able to fill the BC at the attacked CN. In this attack, Eve may simply generate random hash (h(n)) values and HoA addresses, and insert a new pair for each of her 2a messages. The only requirement is that the HoA addresses must be equal in subnet prefix to the prefix of the address used by Eve when Eve is acting as a HA. Otherwise the CN will not reply with a 2b message, and the attack will not be successful (Veigner, 2004).

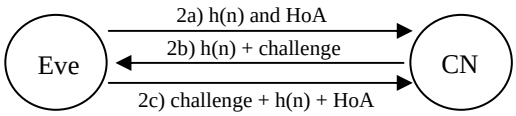

Figure 6: Attacking a CN's binding cache.

HoA	h(n)	CoA	Seq#	Lifetime

Figure 7: The BC at CNs in the ROM protocol.

Even though Eve has to send twice as many messages as the attacked node, Eve may easily carry out her BC flooding attack.

The BC at CNs when using the ROM protocol is shown in figure 7. For each of the MNs that carry out route optimization with the CN, a new row is added to the CN's BC. Each row consists of the following: The mapping from MN's home address (HoA) to its care-of address (CoA), a hash value (h(n)), a sequence number containing highest received sequence number from MN, and finally, remaining lifetime of MN's current binding.

The described BC flooding attack aims to flood the HoA, h(n) and Lifetime columns of the BC. The CoA and Seq# values are only added if a verifiable BU message is received later on (Veigner, 2004).

A MN's BC entry is deleted after 420 seconds if not updated (Veigner, 2004). By re-initiating the ROM protocol, a legitimate MN may update its entry in CN's BC. This solution was chosen for several reasons. The first reason was the fact that there are no known existing one-way hash functions yet. By restricting the valid time of an entry to 420 seconds, and at the same time using a fairly secure hash function, we obtain a one-way hash function for the duration of the 420 seconds. No adversary is able to divert a valid nonce value of an eavesdropped hash value, and is thereby unable to launch a redirecting attack (Veigner, 2004). Due to this 420 seconds validity, a MN must during this period send its BU message to authenticate its current location. Otherwise, a new protocol run is required to update the CN with a new hash value. Another reason for including this validity period in our protocol was to delete unused entries from a CN's BC. As mentioned, a CN may be any node, even a node with limited resources allocating memory for such a BC. Thereby, it is beneficial to delete entries that are not in use.

As a bonus, the described BC flooding attack on the ROM protocol suffers from the deletion of BC

entries. An attacker must now be very efficient, or launch the attack in a distributed manner, to flood the attacked BC within 420 seconds.

When the BC flooding attack on the ROM protocol was discovered, it became in our interest to search for a similar attack feasible on the RR protocol. Studying (Hinden, 2003) gave us the idea of how this could be done. The attack is introduced in Section 4.1 and Section 4.2.

4 THE RETURN ROUTABILITY (RR) PROTOCOL

In this section an overview of the RR protocol is given. RR is the route optimization protocol suggested by the IETF for authenticating a MN's binding update (BU) message sent to a CN. Whenever a MN moves from one subnet to another, it has to initiate route optimization with its CNs. When updating its binding at a CN, MN has to send and receive the messages of figure 8. The message exchange with the CN is carried out subsequent to the BU and BUAck message exchange with the HA. In RR, the message exchange of figure 8 is carried out when MN arrives at its new subnet.

In brief, the MN receives a key-generated token in the HoT message and another key-generated token in the CoT message. When a BU message is finally sent from MN to CN, MN must use its received tokens to make CN confident in MN's authenticity. MN has shown its ability to receive tokens at its two stated addresses (HoA and CoA) via two different routes, and is thereby authenticated by CN (weak authentication). The data from CN may now be routed directly to MN's new CoA address.

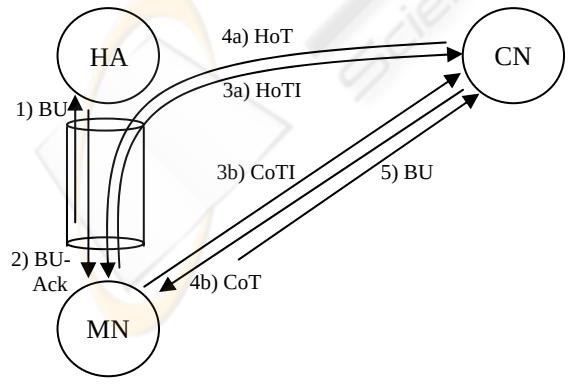

Figure 8: The Return Routability protocol.

We now introduce the RR protocol messages. Message 1 and message 2 are left out. The so-called home routability test and the care-of routability test are the two parts of the RR procedure.

The *home routability test* consists of the HoTI and HoT messages. The *care-of routability test* consists of the CoTI and CoT messages.

$$HoTI = \{HoA, CN, C_h\}$$

The HoTI message comprises the following. The home address (HoA) of the MN is the source address and the CN's address is the destination address. A home init cookie generated by MN is included. This cookie is returned in the response message from CN. MN is now able to match request with response. The HoTI message is reverse tunnelled through MN's HA.

$$CoTI = \{CoA, CN, C_c\}$$

The CoTI message consists of the following. The care-of address (CoA) of the MN appears as the source address. The care-of address is the address used by MN at its foreign subnet. The destination address of the message is the CN's address. A care-of init cookie generated by MN is also included. This cookie must be returned in the response message from CN.

$$HoT = \{CN, HoA, C_h, Token_h, i\}$$

CN generates the HoT message on reception of the HoTI message. This message is sent from CN to MN's HoA address. It is the HA at MN's home subnet that is responsible for redirecting the HoT message to MN when MN is away from its home subnet. On reception of the HoT message, MN uses the C_h cookie to match request with response. MN is now in possession of a key generated token called *home keygen token* ($Token_h$).

$$Token_h = First\ (64, HMAC\_SHA1\ (K_{CN}, (HoA|\ nonce_i|0)))$$

The CN generates the home keygen token by using the first 64 output bits from a MAC function. Input to the MAC function is CN's secret key (K_{CN}) and the concatenation of MN's HoA address, a nonce value and a 0 octet.

The final parameter of the HoT message is the home nonce index (i). MN must later on return this index in its BU message. The CN may thereby remain stateless until the BU message is received. From a list at the CN, containing valid nonce values, the correct nonce value is easily recovered due to this index. The CN may then regenerate the home

keygen token. Both the MN and CN use this token in the generation of their shared *binding management key* (K_{bm}). Another token is also needed in the binding management key generation. This token is sent to the MN in the CoT message.

$$CoT = \{CN, CoA, C_c, Token_c, j\}$$

CN generates a CoT message on reception of the CoTI message. The address of CN is source and the MN's CoA address as destination. This message is sent directly to MN at its current location. The cookie from the CoTI message is included, and a *care-of keygen token* ($Token_c$) is generated quite similar to the $Token_h$.

$$Token_c = First\ (64, HMAC\_SHA1\ (K_{CN}, (CoA| nonce_j|1)))$$

Finally the care-of nonce index (j) is included in the CoT message. Later on this index is returned in the BU message from MN. Thereby helping the stateless CN identifying the nonce value used in the generation of the care-of keygen token ($Token_c$).

The MN is now in possession of both the keygen tokens and may generate a binding management key by hashing the tokens in the following way:

$$K_{bm} = SHA1\ (Token_h|Token_c)$$

Finally the RR procedure is finished. The MN may now generate and send its BU message to the CN.

$$BU = \{CoA, CN, HoA, Seq\#, LT, i, j, MAC_{BU}\}$$

$$MAC_{BU} = First\ (96, HMAC\_SHA1\ (K_{bm}, (CoA| CN|BU)))$$

In the BU message, MN's CoA address is source and the CN's address is destination. The MN's home address (HoA), a sequence number, proposed lifetime for the binding and the nonce indices are all part of the BU message. The HoA and indices are needed by the CN to be able to regenerate the keygen tokens. The CoA is also needed for this purpose. A MAC is finally appended to the BU message.

On reception of the BU message, the CN generates the K_{bm} from its regenerated keygen tokens. By use of the K_{bm}, CN is able to verify the MAC. Whenever a BU message is considered authentic, CN updates its binding cache (BC) with an entry of the MN.

4.1 BC flooding attack on the RR protocol I

In this section we introduce our BC flooding attack on the RR protocol.

In general, whenever there is a cache or buffer that needs to be allocated memory resources, attackers might try to take advantage of it. An attacker may simply fill such storages with random data, resulting in others, non-fraudulent nodes, impossibility of updating the storages with usable information.

An attack with similar outcome as our proposed attack is briefly described in (Nikander, 2005). An attacker sends a spoofed packet to a MN. The packet appears to originate from a CN wanting to initiate communication with the MN. The CN is the attacked node in this scenario. The packet must be sent via the MN's home subnet, i.e. non-optimized routing. On reception of the packet, MN will initiate route optimization with the attacked CN. The protocol will be executed according to the specifications, and an entry of the MN will be added to the CN's BC. The proposed attack (Nikander, 2005) manage to flood the attacked CN's BC only if a sufficient number of entries are added to the BC before to many previously added entries starts expiring. In other words, the cache must be filled to maximum capacity, leaving the attacked node unable to perform route optimization with other MNs. To succeed in its attack, the attacking node must know the addresses of sufficiently many MNs. Unless such a MN is a MN away from its home subnet, the attacker will not succeed in getting the MN to initiate the Return Routability protocol with the attacked node.

However, the described attack is possible against any binding update authentication protocol, but finding sufficiently many MNs to succeed in the attack, might become challenging. Ingress filtering also renders the attack more difficult, since it makes it harder to forge the source address of the spoofed packets. We will therefore introduce a new way of launching BC flooding attacks on the RR protocol, showing that RR as well as our ROM protocol easily may become target of BC flooding attacks.

An IPv6 node may allocate several IPv6 addresses to a single interface (Hinden, 2003). This gave us the idea of how a CN may be victim to a similar BC flooding attack when using the RR protocol, as when using the ROM protocol.

Consider an Eve configuring lots of IPv6 addresses to a single interface. This may be done in an IPv6 stateless address autoconfiguration manner (Thomson, 1998). Eve is now associated with several IPv6 addresses, all with subnet prefixes

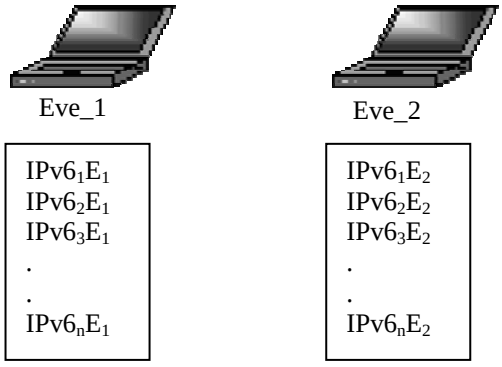

Figure 9: Two Eves, both associated with numerous IPv6 addresses.

equal to the prefix of the subnet where Eve is located.

Finally, by having two Eves at different subnets, both associated with lots of IPv6 addresses as shown in figure 9, the BC flooding attack is possible.

The attack may be launched in the following way. If Eve_1 initiates a home routability test with a victim node as shown in figure 10, i.e. sends a HoTI message to the victim, she will receive a HoT message in reply. The HoT message contains a home keygen token ($Token_h$). To the attacked CN, the source address of the HoTI message seams to be the home address (HoA) of a legitimate MN. But in this attack, the address is actually one of Eve_1's previously configured addresses.

If Eve_2 initiates a care-of routability test she will receive a CoT message, and will thereby also be in possession of a keygen token, not a home keygen token ($Token_h$), but a care-of keygen token ($Token_c$).

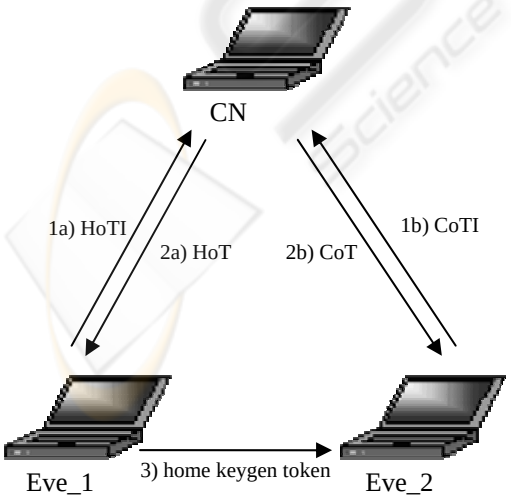

Figure 10: Flooding attack on CN's BC.

Now, if Eve_1 forwards her $Token_h$ to Eve_2, Eve_2 will be capable of sending a verifiable BU message to the attacked CN. The BU message must be generated using the binding management key (K_{bm}). This key is generated by means of the received keygen tokens.

From the attacked CN's point of view, it seams as if a MN with HoA address equal to the address used by Eve_1 has moved to the address used by Eve_2. Whenever a verifiable BU message is received, a mapping from the MN's HoA address to the MN's currently used care-of address (CoA) is added as an entry in the CN's BC. If an entry of this MN already exists, the existing entry is only updated.

The functionality of the RR protocol makes this a perfectly feasible attack, having *two* conspiring Eves flooding a CN's BC. The CoTI-initiating Eve should be responsible for sending the final BU messages. This Eve is associated with the CoA addresses, and the BU messages must originate from these addresses to succeed in the attack. However, the HoTI initiating Eve could also be doing this, but then its source address must be spoofed, appearing to be its conspiring node. Due to ingress filtering this might become challenging. Hence, the CoTI initiating node should generally be sending the BU messages.

In the RR protocol, a CN should generate a new private key (K_{CN}) and nonce value every 30 seconds. The CN should also remember eight of its previously used private keys and nonce values. A private key and nonce value pair is used as a part of the keygen token generation. If every K_{CN} and nonce value pair is employed by CN for the duration of 30 seconds (Johnson, 2004), the issued $Token_h$ and $Token_c$ are valid for the duration of 210–240 seconds from first being issued, depending on where within the 30 seconds time interval the tokens were generated by the CN. To summarize; the tokens are valid, and may be used by the attacking Eves to generate a verifiable BU message, as long as the CN may use the tokens to authenticate the received BU message, i.e. as long as CN has the previously used K_{CN} and nonce value in its memory.

The tokens are independent of each other, and hence, the HoTI and CoTI messages of figure 10 must not necessarily be sent simultaneously. Synchronization of the attacking Eves is hence unnecessary. To succeed in the BC flooding attack, the tokens must be considered authentic by the attacked CN. This is verified by the CN on reception of the BU message.

Initiating the home and care-of routability tests repeatedly, using the previously configured IPv6 addresses, and sending of BU messages as explained, may eventually fill the BC at the attacked

CN. New IPv6 addresses may of course be dynamically configured by the attacking Eves during the attack.

To minimize the effect of different known and unknown attacks, the designers of the RR protocol introduced a 420 seconds durability of the mapping from a MN's HoA address to the MN's CoA address. If not updated, MN's entry in CN's BC is deleted. In addition to making different attacks more complicated, the deletion of BC entries is used as memory management. A CN may delete entries from its BC when not in use.

Due to the removal of BC entries, even an honest MN must initiate the RR procedure and send new BU messages to its CNs at least every 420 seconds. This feature was also used in the design of our ROM protocol. Attackers must now execute their BC flooding attacks within a 420 seconds time interval.

4.2 BC flooding attack on the RR protocol II

IPv6 will continue to use the model from IPv4 (Hinden, 2003); a subnet prefix is associated with one link and multiple subnet prefixes may be assigned to the same link, e.g. an Ethernet. This will ease our flooding attack, reducing the need of two cooperating Eves to launch the attack, to only one Eve. If the subnet where an Eve is located is assigned multiple subnet prefixes, Eve may act as both Eve_1 and Eve_2. Eve may now configure lots of IPv6 addresses using two different subnet prefixes. Eve may then by herself launch the attack of Section 4.1. Of course Eve may have to be a more powerful node in this attack scenario than in the scenario of Section 4.1.

Since every node in MIPv6 may become a CN of a MN, and the MIPv6 protocol is supposed to be a default part of the IPv6 protocol, any IPv6 node may be victim to this BC flooding attack. However, attacking a node that is often used as a CN by other MNs, will be more harmful than attacking a node that is never used, and hence not in need of its BC.

5 CONCLUSION

Return Routability (RR) is the route optimization protocol suggested by the IETF (Johnson, 2004). RR is used to authenticate binding updates sent from mobile nodes (MNs) to corresponding nodes (CNs). Our ROM protocol (Veigner, 2004) intends to make MIPv6 route optimization more seamless than RR manage; in other words, to speed up the procedure and at the same time provide similar security characteristics. The importance of the protocol being seamless is the fact that route optimizations are often carried out during a MN's handover from one subnet to another.

This paper focuses on flooding attacks on the binding cache (BC) at CNs, and shows to which extent the RR protocol as well as our ROM protocol is vulnerable to such attacks.

Certain countermeasures have been suggested. The 420 seconds durability of BC entries is already included in both protocols. Nevertheless, the BC flooding attack discovered on the IETF suggested RR protocol is important to point out. Another countermeasure is to keep the number of entries allowed in a BC low. This is not necessarily a good solution, making DoS attacks easier to carry out by means of BC flooding attacks. Strong authentication was also considered, but the solution has a major disadvantage in scalability due to the lack of a global PKI. Use of asymmetric cryptography would also be a very CPU-consuming feature; resulting in increased DoS attack vulnerabilities.

As we all know, bandwidth in mobile and wireless networks is unpredictable and often low. In comparison to RR, the main benefit of the ROM protocol is the reduction of messaging when re-establishing route optimization from a new subnet.

It is important that we understand all the threats the new technology creates before a possible deployment.

REFERENCES

Aura, T., 2002. *Mobile IPv6 Security*, Cambridge Security Protocols Workshop.

Deng, R. H., Zhou, J., Bao, F., 2002. *Defending Against Redirect Attacks in Mobile IP*, Proceedings of the 9th ACM conference on Computer and communications security.

Hinden, R., Deering, S., 2003. *Internet Protocol Version 6 (IPv6) Addressing Architecture*, IETF RFC 3513.

Johnson, D., Percins, C., Arkko, J., 2004. *Mobility Support in IPv6*, IETF RFC 3775.

Nikander, P., Arrko, J., Aura, T., Montenegro, G., Nordmark, E., 2005. *Mobile IP version 6 Route Optimization Security Design Background*, IETF Internet-draft.

Thomson, S., Narten, T., 1998. *IPv6 Stateless Address Autoconfiguration*, IETF RFC 2462.

Veigner, C., Rong, C., 2004. *A new Route Optimization protocol for Mobile IPv6 (ROM)*, International Computer symposium 2004, Taipei.