Denial of Service Prevention in Satellite Networks

Melek Önen, Refik Molva
Institut Eurécom
Sophia-Antipolis - France
Email: {Melek.Onen,Refik.Molva}@eurecom.fr

***Abstract*— Networks integrating satellite shared access, such as DVB-S/RCS, are particularly exposed to Denial of Service (DoS) attacks through which a satellite terminal can maliciously use the Network Control Center (NCC) resources by submitting a high number of bogus requests. Anti-clogging techniques used in terrestrial mesh networks to thwart DoS attacks fall short of solving DoS problems in the case of satellite networks due to the inherent broadcast capability of the communication medium. We suggest a generic mechanism to prevent DoS attacks in control plane protocols. The principle behind this mechanism is an efficient identification technique that allows the satellite servers to quickly discard bogus requests. Thanks to this technique, intruders controlling the total bandwidth of the up-link can only consume a very small fraction of the NCC's resources.**

I. Introduction

The shared access capabilities provided by satellite systems such as DVB-S/RCS [1] or UDLR [2] raise serious security concerns. An important issue is the ease at which unauthorized users can get access to satellite services. A straightforward countermeasure consists of access control based on user authentication. Unfortunately, while solving the access control problem, strong authentication brings up a new vulnerability that is denial of service (DoS). Aiming at preventing legitimate users from accessing the service provided by the satellite system, a simple solution using access control based on strong authentication procedures embedded in the satellite access control system would still allow intruders to perpetrate DoS attacks by submitting a large number of bogus requests messages. Since the authentication of each request involves CPU intensive cryptographic operations and allocation of memory space required to store protocol state information, the screening of a large number of bogus requests thus results in high CPU utilization and occupation of large memory segments at the expense of potential legitimate requests that are delayed or simply discarded by the server.

There exist some authentication protocols proposed for meshed terrestrial networks including the protection against DoS attacks based on the anti-clogging technique [3]. This technique introduces a weak authentication phase prior to the actual authentication in order to screen out authentication requests sent by an intruder with a bogus source address. During this weak authentication phase, the server replies to the claimed source address of each request providing some information that would allow the recipient to further proceed with the actual authentication phase. Since requests resulting from a DoS attack bear a bogus source address, the corresponding weak authentication replies sent by the server will not reach the intruder preventing the latter from being able to further consume server resources with the actual authentication phase.

This work is partially supported by Alcatel Space Industries, Toulouse, France.

Such DoS prevention technique is not suitable to the satellite network environment because the screening of bogus requests by the weak authentication mechanism will not be effective due to the inherent broadcast capability of the satellite segment. Due to the broadcast medium, intruders can receive all traffic including the weak authentication replies destined to bogus addresses from DoS attacks, thus eliminating the benefit of the anti-clogging technique.

Therefore, we propose an efficient identification protocol that allows the satellite servers to quickly discard bogus requests without even sending replies to the claimed address of each request to get the assurance of the legitimacy of the request. Our mechanism is based on the use of strong authentication techniques for each message where the consumption of the server's resources during the verification phase is optimized.

We first present existing DoS prevention techniques designed for terrestrial networks and show their shortcomings in the satellite environment. We then describe a first basic identification protocol requiring a strong synchronization mechanism between the satellite terminals and the Network Control Center (NCC). We further enhance this protocol in an improved version that only requires loose clock synchronization between the NCC and the terminals. We then evaluate the effectiveness of our DoS prevention mechanism using a simulation of DoS attacks in various settings in terms of NCC CPU power. The results of the simulations show that even a powerful attacker that controls the total up-link bandwidth can only consume a minor fraction of NCC resources.

II. Problem Statement

A. Definition of Denial of Service Attacks

Denial of Service attacks aim at preventing legitimate users from accessing a service they are authorized to access in normal circumstances. CERT classifies three kind of DoS attacks in [4] :

- Consumption of scarce, limited, or non renewable resources;
- Destruction or alteration of configuration information;
- Physical destruction or alteration for network components.

Destroying or altering a configuration information is only possible when the network or the computer is not properly configured. In these cases, an intruder may prevent the legitimate user from using the network or its computer. We suppose that the satellite system and its components are properly configured and protected physically against the unauthorized access. Consequently, the last two kinds of attacks will not be under the scope of this paper and we focus on the first type of attacks where the target of an attacker is the victim's principle resources which are : the memory and the disk space, the network bandwidth and the CPU time.

In a typical attack, the intruder may be able to consume a significant portion of the memory and the disk space at the target by sending some special packets based on the mandatory memory allocation caused by the processing of each packet. The "SYN flood attack" described in [5], [6] is a typical example of a memory consumption attack. This kind of attacks may also affect CPU usage if the verification of packets involves computationally intensive operations. Moreover, the intruder may also consume the available bandwidth entirely by generating a large number of faked packets.

Since the rate of requests originating from a legitimate entity lays below a limit that is acceptable for the server, in order to justify an increased rate, DoS attacks persistently emulate several sources by either setting bogus source addresses in requests generated by a single intruder, by having recourse to several sources as distributed intruders, or both. The first approach used by DoS prevention methods therefore consists of screening requests with bogus source identification. Even though a perfect measure to that effect, a straightforward implementation of data origin authentication techniques unfortunately does not solve the DoS problem. As a computing intensive operation based on cryptographic algorithms, like digital signature or encryption, the authentication of the requests becomes the new target for the DoS attacks.

B. Denial of Service Prevention in Terrestrial Networks

IPSec [7], has defined a framework for key exchange and negotiation of security associations (SA), called ISAKMP [8]. This protocol allows peer entities to select and negotiate the SA with the incorporation of a mechanism to counter DoS attacks. ISAKMP is based on the anti-clogging technique [3] where an exchange of weak authentication information called cookie occurs at the beginning of each client-server connection before initiating any resource-intensive verification.

A cookie is generated by each communicating entity and has some special characteristics [3] fostering a fast and efficient exchange :

- It is generated based on a local secret known only by its generator;
- It depends on the addresses of the communicating entities;
- It is verifiable by the generator;
- Its generation and verification are efficient in CPU and memory consumption.

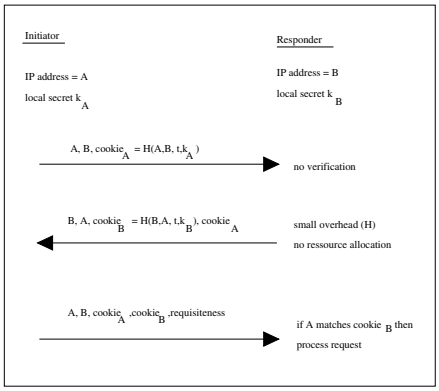

Fig. 1. The principle of the anti-clogging technique

Anti-clogging is carried by the cookie exchange protocol as depicted in figure 1. This exchange allows the verification of the client's presence at the claimed IP address. The client initially sends a request to the server with its cookie. Upon the receipt of this request, the server sends without any verification, another cookie based on the claimed IP address of the client retrieved from the request. Because the server's reply including server's cookie is sent to the claimed client address, an intruder using a bogus IP address cannot get the server's cookie. Consequently, in case of a request with a bogus source address, the server will not receive any third message including the client's and the server's cookies, hence the DoS attack intended by the bogus request will not succeed. Only requests originating from legitimate parties will thus reach the server. Moreover, the computation of the cookie by the server is based on a simple hash function requiring low CPU usage in comparison with CPU-intensive strong authentication and key generation operations and no resource reservation takes place before the completion of the successful cookie exchange [9]. Each ISAKMP message contains the pair of cookies generated by the initiator and the responder based on the anti-clogging technique.

C. Requirements of the Satellite Network

The previously depicted anti-clogging technique is ineffective in the satellite environment for two reasons : the broadcast nature of communications and the end-to-end latency.

Due to the inherent broadcast nature of satellite communications, each terminal can receive all traffic from the servers. In this context, the anti-clogging technique based on the cookie exchange would be ineffective because the cookies generated by the server and intended for sources impersonated by the intruder would still be received by the intruder who will then be able to reply with the expected weak authentication messages successfully masquerading as several different sources. The screening of bogus source addresses that is perfectly suitable to mesh networks would thus be totally ineffective in case of broadcast media like the satellite segment.

Moreover, satellite networks inherently experience long delays (end-to-end latency between two satellite terminals is approximatively 300 ms [10]). The additional delay that

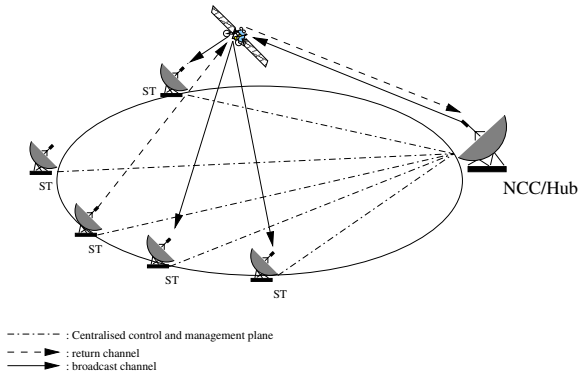

Fig. 2. A Satellite System Architecture

would be caused by including an anti-clogging phase with three additional messages in the satellite protocols would thus be a significant drawback that is unacceptable for many applications.

An authentication protocol protecting a satellite network from DoS attacks should not only be secure against masquerade, but also be efficient in terms of resource consumption and number of flows.

III. The Basic Protocol

In order to introduce our DoS prevention solution for satellite networks, we first briefly sketch a satellite system architecture. The solution is then presented as a protocol run by the components of this architecture.

A. Environment

In a typical satellite system architecture, network control and management are centralized and under the control of the Network Control Center (NCC) which is usually co-located with the Hub-Station responsible for the data-plane and some management functions such as satellite system address management.

A satellite terminal (ST), sends control messages such as connection requests to the NCC. These messages can be transmitted via satellite using DVB-RCS or data transfer protocols such as UDLR.

B. Description of the Basic Protocol

For the basic protocol, we assume that there is a tight clock synchronization between the NCC and the terminals. This requirement will be released with the improved version of the protocol described in section 5.

In both versions of the protocol, each satellite terminal is assigned with a unique identity, such as the Media Access Control (MAC) address of the terminal. Initially, every terminal shares a secret key with the NCC which is precomputed by the NCC. Moreover, the NCC divides the time in fixed intervals T_j during which it reliably broadcasts a different nonce N_j for the purpose of replay detection.

Unlike DoS prevention protocols designed for terrestrial mesh networks, the proposed identification protocol requires

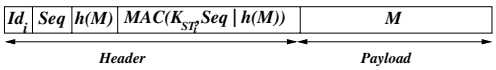

Fig. 3. The structure of a message sent by ST_i at time T_j

only one message flow from the satellite terminal to the NCC. The figure 3 depicts the structure of a request message based on the following notation :

- Id_i denotes the identity of the satellite terminal ST_i which is unique for each ST;
- Seq denotes the sequence number of the message sent by a ST, which is initially equal to the nonce N_j of the current interval T_j;
- h denotes a cryptographic hash function such as MD5 [11] or SHA1 [12] and MAC denotes a cryptographic keyed hash function such as HMAC [13];
- K_{ST_i} denotes the secret key shared between ST_i and the NCC defined as $K_{ST_i} = MAC(K_{NCC}, Idi)$ where K_{NCC} denotes a secret key known only by the NCC;
- $a|b$ denotes the concatenation of a and b;
- Finally, M denotes the payload of the request message sent by a ST.

Upon receipt of a request sent by a satellite terminal ST_i during the interval T_j, the NCC first verifies that the sequence number Seq is equal to the actual nonce N_j. It then computes the secret shared key K_{ST_i}, using the identity Id_i retrieved from the header of the request message. In order to verify the authenticity of the request, the NCC computes the value $MAC(K_{ST_i}, Seq|h(M))$ using the $h(M)$ value retrieved from the header. If this computed value and the one retrieved from the header match, the NCC needs to further verify the integrity of the message by evaluating the hash value of the payload and comparing the result with the corresponding value received with the message.

This two-step verification process allows for fast discarding of control messages with obvious inconsistencies. Once both verification steps of a request sent by a ST end with success, the NCC processes the request and allocate necessary resources for it. However, in this protocol, we supposed that an intruder could not send replayed messages eavesdropped within the same interval. In section 5, an improved protocol will be defined by taking into consideration this type of replay attacks.

IV. Security Analysis of the Protocol

Based on the secrecy of K_{ST_i} shared by the corresponding ST_i and the NCC, an intruder cannot generate a valid request. We then turn to the analysis of successful replays by an intruder. First, we consider the replay of legitimate requests from an interval in subsequent intervals by the intruder. This type of replays will be easily detected by the NCC based on the difference between the current nonce value and the value included in the header of the replayed request. Furthermore, attacks aiming at substituting or tampering with the payload of a legitimate request cannot succeed since the modification or

substitution of the payload will be detected during the second phase of the verification.

We now discuss the case where the generation and replay of a request are done within the same interval. For a satellite network, it is practically impossible to intercept any message from the up-link. Therefore, an intruder can intercept a legitimate terminal's request only from the down-link, that is, after an end-to-end latency for a GEO-satellite system which is approximatively equal to 300ms [10]. Thus, if the interval is set to a value less than the end-to-end latency and a different nonce is used in each interval, the intruder cannot successfully perpetrate this type of replay.

V. Improvement of the Protocol

In this section, we propose to improve the basic protocol by introducing a stateful verification mechanism that allows to relax the synchronization requirement. The improvement is based on the following idea : by relaxing the synchronization between the NCC and the terminals, some replays occurring under race conditions would go undetected; in order to detect those replays the NCC will keep some state information about successful authentication attempts occurring in each interval.

For each interval T_j :

Preliminaries :

Define and reliably broadcast N_j
Define an initially empty table which arguments for each authenticated ST_i are :
the identity of the ST_i, Id_i
its shared key K_{ST_i} and,
the last received sequence number Seq_i

The verification of a request received during interval T_j:

Request = { Id_i, Seq, $h(M)$, $MAC(K_{ST_i}, Seq|h(M))$, M }
if $Request.Id_i \notin table$
if $Request.Seq \neq N_j$
REJECT
else
compute $K_{ST_i} = MAC(K_{NCC}, Id_i)$
compute $MAC(K_{ST_i}, Seq|h(M))$
if $MAC(K_{ST_i}, Seq|h(M)) \neq Request.MAC(K_{ST_i}, Seq|h(M))$
REJECT
else
compute $h(M)$
if $h(M) \neq Request.h(M)$
REJECT
else
add { Id_i , K_{ST_i}, Seq } to the table
ACCEPT
if $Request.Idi \in table$
if the value $Seq_i \leq Request.Seq$
REJECT
else
compute $MAC(K_{ST_i}, Seq|h(M))$
if $MAC(K_{ST_i}, Seq|h(M)) \neq Request.MAC(K_{ST_i}, Seq|h(M))$
REJECT
else
compute $h(M)$
$h(M) \neq Request.h(M)$
REJECT
else
replace $Seq_i = Request.Seq$
ACCEPT

Fig. 4. Verification process of a request with the improved protocol

At the beginning of a verification interval with a new defined nonce N_j, when the NCC receives and successfully verifies the authenticity of the first request sent from a legitimate ST, it keeps some information about this request in a table reset at the beginning of each interval. The arguments of the table for each different satellite terminal authenticated within interval T_j are : the ST's identity, Id_i, its secret key, K_{ST_i}, and the sequence number of the message which has to be initially equal to the actual nonce N_j.

The difference between our previous basic protocol and this improved one, is that the ST will increment the sequence number Seq initially equal to the nonce of the period for every new request within the valid interval. Thus, when the NCC will receive another request from a terminal which previously has been added to the table, it will also verify that the sequence number sent by the ST is greater than the one in the table, and replace the value with the new one if the message is authenticated. Since an intruder cannot generate a valid request, it therefore will not be able to make a replay attack within a same interval where requests have been eavesdropped, because the sequence number changes for each message.

The new version also optimizes the CPU consumption because the NCC won't need to compute the shared key K_{ST_i} if this value is already present in the table defined below. We assume also that the NCC can allocate necessary resources for the dynamic table re-initialized at the beginning of each interval and that searching if a key is already existing in the table is much more efficient than computing its value using MAC algorithms.

This protocol presents also all the secure properties offered by its old version. Thus, if the NCC allows an extension of the duration of one interval, with the implementation of the improved protocol, generation and replay attacks within a same interval would not be possible.

VI. Evaluation of the Protocol

In order to evaluate the possibility of DoS attacks with our protocol, we consider the cost memory and CPU usage due to an attack performed by an intruder taking advantage of all available bandwidth. Since the NCC only keeps track of authenticated terminals in an array that is reset at each interval, the resulting usage is optimal and cannot be exploited by an intruder.

As to the evaluation of the CPU usage, we first compare the cost of computing the MAC over the entire message and the MAC over the hash value of the message in order to validate the necessity of the first step of the verification of the proposed protocol. The table I shows the number of messages verified in one second and compares the evaluation of the two MAC values. CPU consumption has been evaluated with OPEN-SSL [14], where the HMAC and MD5 cryptographic functions are respectively used for MAC and hash functions. We consider that the message is 500 bytes long. With these assumptions, we see that for a Pentium 4 2,4 GHZ with 1GB of RAM, 326435 packets are verified in one second in the case of the verification of $h(M)$ as opposed to 206289 packets in the case where the input of the MAC function is the message M itself. Thus, the first step of verification increases the number of verified packets by 58% .

Moreover, we consider a scenario whereby the NCC is under a DoS attack by an intruder that uses the total amount of the bandwidth set up to 10 Mbps. Such an intruder can send at

TABLE I

CPU USAGE OF THE NCC PER SECOND FOR THE COMPUTE OF HMAC OVER M AND $h(M)$

Workstation Properties	Number of verified packets per second	
	$MAC(h(M))$	$MAC(M)$
sparc v9 300Mhz RAM = 256MB	49783	26022
sparc v9 440Mhz RAM = 640MB	57846	33055
Pentium 3 1Ghz RAM = 896MB	141312	103280
Pentium 4 2,4Ghz RAM = 1GB	326435	206289

most 2621 packets per second based on these assumptions. Table II shows the percentage of CPU usage by the NCC as required for message authentication. One can note that the CPU usage due to the computation of a MAC over a hash value is approximately 50% more efficient then the one over the message itself.

TABLE II

CPU USAGE PER SECOND OF THE NCC FOR THE VERIFICATION IN CASE OF A DOS ATTACK

Workstation Properties	CPU usage per second for verification	
	$MAC(M)$	$MAC(h(M))$
sparc v9 300 Mhz RAM = 256MB	10,07%	5,26%
sparc v9 440 Mhz RAM = 640MB	7,93%	4,53%
Pentium 3 1Ghz RAM = 896MB	2,54%	1,85%
Pentium 4 2,4GHz RAM = 1GB	1,27%	0,80%

VII. CONCLUSION

Denial of Service is becoming an important concern for networks. The approach taken for mesh terrestrial networks has been to use the anti-clogging technique based on an exchange of cookies in order to screen out requests with bogus source addresses by sending replies including a cookie to the claimed address. This technique however turns out to be inefficient to thwart DoS attacks in satellite networks because of the inherent broadcast capability of the satellite system and thus continue to send bogus messages with correct cookies, the terrestrial solutions are not suitable to satellite networks.

We proposed two versions of an efficient identification protocol protecting against DoS attacks for centralized control plane protocols in satellite networks. Since the basic protocol requires a tight clock synchronization between the satellite terminals and the NCC, we proposed an improved protocol to relax the synchronization assuming that the NCC keeps some state information about successful authentication attempts in each interval. Using this improved technique, we showed that intruders controlling the total bandwidth of the satellite link can only use a very small percentage of the NCC's CPU without requiring a strong synchronization mechanism. However, our protocol does not prevent attacks whereby a satellite terminal can monopolize the common signaling channel for a very long period and thus no other terminal can communicate with the NCC. Such attacks can be avoided only by identifying the intruder and preventing its physical access.

Future work should focus on the analysis of weaknesses of the DVB-RCS standard authentication protocols and the evaluation of the effectiveness of our protocol in this context.

ACKNOWLEDGMENT

The authors would like to thank Sébastien Josset for the valuable insights he gave on satellite networks and their security requirements.

REFERENCES

[1] ETSI EN 301 790, "Digital Video Broadcasting (DVB), Interaction Channel for Satellite Distribution Systems," .
[2] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, "A link-layer tunneling mechanism for unidirectional links," RFC 3077, March 2001.
[3] P. Karn and W. Simpson, "Photuris : Session-key management protocol," RFC 2522, march 1999.
[4] CERT Coordination Center, "Denial of Service attacks," February 1999, http://www.cert.org/tech_tips/denial_of_service.html.
[5] CERT Coordination Center, "CERT Advisory ca-1996-21 TCP SYN Flooding and IP Spoofing attacks," November 2000, http://www.cert.org/advisories/CA-1996-21.html.
[6] C. Meadows, "A formal framework and evaluation method for network denial of servic e," in *PCSFW : Proceedings of the 12th Computer Security Foundations Wor kshop*. 1999, IEEE Computer Society Press.
[7] R. Atkinson, "Security Architecture for the Internet Protocol," RFC 1825, August 1995.
[8] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, November 1998.
[9] R. Molva, "Internet Security Architecture," *Computer Networks : The International Journal of Computer and Tele communications Networking*, vol. 31, no. 9, pp. 787–804, 1999.
[10] T. Henderson, *Networking over Next-Generation Satellite Systems*, Ph.D. thesis, University of California at Berkeley, 1999.
[11] R.L. Rivest, "The MD5 message-digest algorithm," April 1992.
[12] National Institute of Standards and Technology, "Secure hash standard," 1995.
[13] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC : Keyed Hashing for Message Authentication," RFC 2104, February 1997.
[14] "The OpenSSL Project," http://www.openssl.org.