Poster / Extended Abstract:

Traffic Flow Confidentiality Enhancements in IPsec: Design and Preliminary Implementation

Simone Teofili, PhD Student, Univ. Roma Tor Vergata, Italy
Fabrizio Formisano, Researcher, CNIT, Italy
Csaba Kiraly, Researcher, CNIT / Univ. Trento, Italy
Giuseppe Bianchi, Full Professor, Univ. Roma Tor Vergata, Italy

MOTIVATION

Traditional communication security focuses on protecting the delivered contents through strong encryption means. However, extensive literature work demonstrates that encryption alone is insufficient to protect confidentiality. The statistical pattern of the traffic generated in a communication carries plenty of information, which can be maliciously gathered through specially devised attacks. By collecting and correlating statistics such as packet size and inter-arrival times (typically left unaltered by most encryption procedures), a third party may be capable of disclosing information such as the application layer protocols and services employed [1,2,3], the physical devices employed [4], or even specific information related to the delivered contents, most notable being the case of attacks against passwords transmitted over encrypted sessions [5,6].

To duly protect the privacy of the users, further mechanisms are needed in addition to encryption. These are frequently referred to as "Traffic Flow Confidentiality" (TFC) mechanisms. Most of the proposed TFC solutions [7,8,9] are however based on custom frameworks which hardly fit with widely deployed communication security protocols such as IPsec and TLS. Moreover, to provide an effective information hiding, TFC mechanisms should be deployed in conjunction with network-wise anonymization approaches, such as Mix-like protocols [10,11,12], devised to mask further information such as chosen routes, involved endpoints, etc. It would be natural to think to TFC as mechanisms provided by a point-to-point (i.e. per-hop) underlying standard protocol (IPsec being a natural candidate), upon which many different network-wise Mix-like protocols are free to develop their own anonymous routing logic. However, to date, every Mix-like protocol is forced to re-develop from scratch its own TFC suite due to the missing support (e.g. in TLS) or lack of satisfactory TFC support (e.g. in IPsec) in the standard security protocols.

CONTRIBUTION

IPsec is the only widely deployed security protocol which has partially tackled the issue of supporting TFC mechanisms. The latest specification of the IPsec Encapsulated Security Payload (ESPv3) protocol [13] introduces limited TFC functionalities in terms of i) partial support for packet padding besides the traditional 255 bytes limit, and ii) dummy packet detection and discarding at the receiver side. However, the IPsec TFC support is not fully satisfactory. For instance, no traffic shaping is accounted for, and no padding extension is possible for encapsulated protocols which lack of an explicit “size” field (a notable case being TCP when carried in transport mode over IPsec). Moreover, the latest IPsec specification does not attempt to specify any (albeit simple) Application Programming Interface to manage TFC mechanisms, leaving their control completely up to the implementations. Finally, the IPsec working group has explicitly chosen not to tackle network-wise issues such as support of Mix-like networks. This makes hard to reuse the native IPsec TFC mechanisms, deployed for an IPsec link (namely, a Security Association - SA) as primitive services for an overlying network-wise Mix-like protocol, and hence forces Mix-like protocol developers to re-design their own TFC mechanisms.

Goal of our work is to propose IPsec enhancements devised to overcome these issues. Specifically, we propose a TFC protocol developed as an extension of IPsec/ESP, specifically as upper sub-layer of ESP. Our TFC protocol is targeted to provide, on one side, an effective support for a variety of TFC mechanisms, and on the other side is devised to simplify the adoption of IPsec (plus our TFC enhancements) as lower layer for Mix-like current and future protocols (and specifically providing per-logical-link TFC mechanisms that overlay network-wise Mix-like frameworks might flexibly exploit and manage).

TFC SUB-LAYER DESIGN

We propose to develop TFC as an upper sub-layer of the current ESP specification, thus maintaining backward compatibility with traditional IPsec implementations. With reference to Figure 1, which depicts an IPsec/ESP packet, the data contained within the ESP payload (i.e. that included between the ESP header and trailer) are further wrapped in a TFC header which provides a further level of indirection to manage TFC tools. A 4 bytes inner TFC header carries three fields: i) Next Header, ii) Type of Confidentiality Treatment (TOCT), and iii) Padding length.

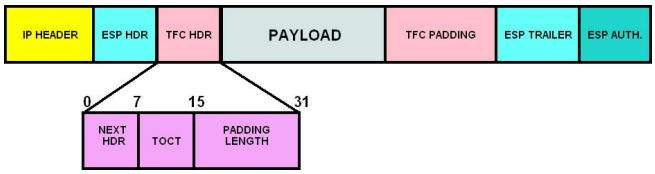


Figure 1 – TFC header

The role of the TFC padding field is straightforward, as it allows to apply arbitrary padding irrespective of the specific protocol carried within IPsec/ESP, thus overcoming the limits of ESPv3. Similarly, the next header field has the obvious goal of specifying the content (protocol) of the TFC frame. This can be either a IANA standard protocol, a dummy packet, or a custom mix-like protocol (eventually addressed through suitable extension headers to improve flexibility).

The main conceptual novelty is the introduction of the TOCT (Type of Confidentiality Treatment) field. Different codes are devised for different TFC treatments. For example, a much shorter random forwarding delay should be applied to real time packets rather than non real time ones. Packet clustering could be enabled or disabled, and so on. This field has the goal to convey, to the receiving side of the IPsec association, the type of TFC treatment that the delivered frame is entitled to receive on an eventual next hop, and is specifically devised with in mind a network-wise multi-hop scenario composed of several IPsec links suitably managed by an upper layer Mix-like protocol. Since overlay mix-like protocols typically deliver encrypted frames (e.g. onion routing), it is necessary to explicitly distinguish the type of packets which may be subject of a different TFC treatment before the inner protocol encryption applies. The TOCT addition allows the upper Mix-like protocol to first classify packets on the basis of their expected treatment, and then deliver the packet within a network composed of several IPsec hops, with the guarantee that a differentiated TFC treatment will be enforced on differently labelled

packets (the reader familiar with the Differentiated Services QoS framework proposed in the late 90's will find many conceptual similarities with it, although for a completely different goal).

IMPLEMENTATION

We have developed a preliminary version of the TFC protocol and specifically implemented a TFC control logic capable of handling three TFC mechanisms:

- Dummy Traffic generation and discarding
- Packet padding
- traffic temporal re-shaping

Our TFC implementation is completely developed inside the Linux Kernel 2.6 (contrarily to other TFC works that typically rely on user-space implementation and thus cannot be integrated in the IPsec Linux protocol stack). Being developed as a sub-layer, the TFC protocol takes advantage of all the existing ESP functionalities (confidentiality, data integrity and authentication, as well as Security Association and policy management).

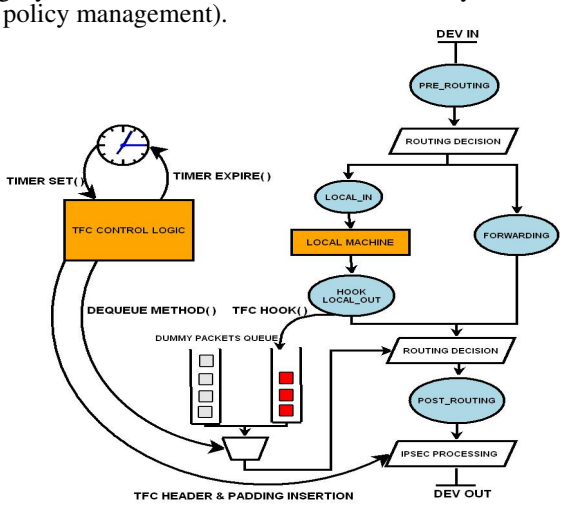


Figure 2 – TFC Kernel Implementation

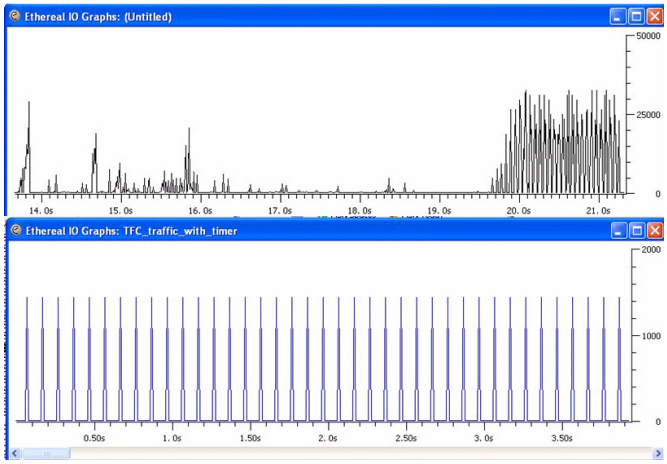
Our implementation is based on the scheme sketched in figure 2. A packet queue is associated to an ESP SA (security association), and initialized when the SA is created. Data packets are intercepted through a custom-made TFC "hook" function. An hook is a well defined point that a packet has to cross in its way to the driver through the IP stack. Multiple functions can be registered to manipulate, discard or make other operations on a packet when this arrives to an hook (example of hooks are PRE_ROUTING, LOCAL_IN, FORWARDING). Our hook function is registered on the LOCAL_OUT hook. If no IPsec transformation or no ESP SA is found for that packet, we leave it unmodified; otherwise the packet is stolen from its normal processing in the Linux kernel and inserted into the appropriate queue.

A TFC control logic module is periodically waken up by a timer. When the timer expires, the TFC module schedules the next timer and sends a packet, duly padded and wrapped into its TFC header, from the head of the packet queue back to the IP kernel stack for transmission. By suitably modifying this timer, different forwarding delays can be added. If no packet is present, the TFC module can take the decision of sending a dummy packet (coded with a next header field set to 59, to conform with RFC 4303). Since our TFC queue is situated before IPsec encryption, dummy packets are sequentially encrypted similarly to data packets, and thus subject to an identical processing time. This prevents an attacker from identifying dummy packets based on different inter-arrival time of the packets.

DEMONSTRATION

The TFC logic currently implemented is very simple, for pure demonstrative purposes. Packets pertaining to a security association are padded, shaped, and complemented with dummy packets, to generate a constant rate of evenly sized packets. Figure 3 reports ethereal captured traces which show how a random mixture of ICMP, HTTP and SSH traffic tunneled into a same IPsec ESP/TFC association are properly assembled and CBR-shaped with even packet sizes (many of them being dummy packets).

Ongoing work is planned to leverage on this initial implementation and aims at tackling two main directions: i) design of algorithms adaptive to the level of congestion and devised to generate non-CBR high entropy traffic are under current study, and ii) demonstrate the multi-hop operation and the TOCT effectiveness by integrating the proposed IPsec ESP/TFC extension into an anonymous routing protocol.



REFERENCES

[1] A. Hintz, "Fingerprinting Websites Using Traffic Analysis Privacy Enhancing Technologies", PET 2002, S. Francisco, USA, April 2002

[2] G. D. Bissias, M. Liberatore, D. Jensen, B. N. Levine, "Privacy Vulnerabilities in Encrypted HTTP Streams", PET 2005, Cavtat, Croatia, May 30-June 1, 2005.

[3] M. Crotti, F. Gringoli, P. Pelosato, L. Salgarelli, "A statistical approach to IP-level classification of network traffic", the 2006 IEEE International Conference on Communications, 11-15 Jun. 2006.

[4] T. Kohno, A. Broido, K. C. Claffy. "Remote physical device fingerprinting", in IEEE Symposium on Security and Privacy, pages 211–225. IEEE Computer Society, 2005.

[5] D. X. Song, D. Wagner, X. Tian, "Timing analysis of keystrokes and timing attacks on SSH", 10th USENIX Security Symposium, 2001.

[6] B. Canvel, A. Hiltgen, S.Vaudenay, M. Vuagnoux, "Password Interception in a SSL/TLS Channel", CRYPTO2003, Aug 2003, Santa Barbara, USA

[7] Y. Guan, X. Fu, D. Xuan, P. Shenoy, R. Bettati, W. Zhao, "NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications", IEEE Trans. on System, Man, and Cybernetics, 2001.

[8] K. Streff, A. Rajagopalan, X. Fu, "ABC: Adaptive Bank-Transaction Camouflaging Systems", ICIW 2006

[9] B. Timmerman, "Secure Dymanic Adaptive Traffic Masking", in New security paradigms workshop, 1999.

[10] G. Danezis, R. Dingledine, N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol", proceedings of the 2003 IEEE Symposium on Security and Privacy, May 2003.

[11] R. Dingledine N. Mathewson, P. Syverson, "Tor: The Second-Generation Onion Router", 13th USENIX Security Symp. Aug 2004.

[12] M. J. Freedman, R. Morris, "Tarzan: a Peer-to-Peer Anonymizing Network Layer", ACM Conf. on Computer and Communications Security (CCS 2002), Washington, DC, November 2002.

[13] S. Kent, "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

For further information please visit:
http://www.ist-discreet.org/tfc.html