Too Big or Too Small? The PTB-PTS ICMP-based Attack against IPsec Gateways

Vincent Roca, Saikou Fall

IPsec Maintenance and Evolution (IPSECME)
Internet-Draft
Intended status: Informational
Expires: January 7, 2016

V. Roca
S. Fall
INRIA

July 6, 2015

Too Big or Too Small? The PTB-PTS ICMP-based Attack against IPsec Gateways

draft-roca-ipsecme-ptb-pts-attack-00

Abstract

This document introduces the "Packet Too Big"-"Packet Too Small" Internet Control Message Protocol (ICMP) based attack against IPsec gateways. We explain how an attacker having eavesdropping and packet injection capabilities, from the unsecure network where he only sees encrypted packets, can force a gateway to reduce the Path Maximum Transmission Unit (PMTU) of an IPsec tunnel to the minimum, which can trigger severe issues for the hosts behind this gateway: with a Linux host, depending on the PMTU discovery algorithm in use (i.e., PMTUd versus PLPMTUd) and protocol (TCP versus UDP), the attack either creates a Denial of Service or major performance penalties. This attack highlights two fundamental problems, namely: (1) the impossibility to distinguish legitimate from illegitimate ICMP packets coming from the untrusted network, and (2) the contradictions in the way Path MTU is managed by some end hosts when this Path MTU is below the minimum packet size any link should support because of the IPsec encapsulation.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2016.

Table of Contents

1. Introduction

IPsec interacts with the Internet Control Message Protocol (ICMP). A first goal of ICMP is to exchange control and error messages, like packet processing error notifications. But ICMP is also involved in several functionalities and in particular the Path Maximum Transmission Unit discovery (PMTUd) mechanism [RFC1191], whose goal is to find the maximum packet size on a path that avoids packet fragmentation. Such a mechanism is essential from a performance point of view: if a packet is too large, its fragmentation and reassembly will negatively impact performance. At the other extreme, if a packet is significantly smaller than the maximum size permitted throughout the path, it will also negatively impact performance. Assessing the correct packet size on a path is therefore essential.

But ICMP is also known to be a cause of attacks and therefore there is an incentive for a network administrator to filter out these packets (see [Jacquin12] for a detailed analysis of the situation). A balance is therefore required between these contradictory objectives, and it is recognized that only a subset of ICMP packets should be considered by IPsec gateways. In this document we assume that the target IPsec gateway accepts and processes the ICMP "Destination unreachable"/"Fragmentation needed" (with IPv4) or ICMPv6 "Packet Too Big"/"Fragmentation needed" (with IPv6) packets coming from the unsecure network. For simplification purposes, the term "ICMP PTB" will be used throughout this document to denote either of these ICMP packets.

The PTB-PTS attack is carried out from the untrusted network, and through the IPsec gateway, the attack targets hosts in the trusted network, behind the gateway. We assume the attacker can eavesdrop and inject traffic on the untrusted network Section 4, i.e., we assume the attacker is on the path followed by the tunnel (which is trivial in case of an unsecure WiFi network). A single ICMP PTB packet is sufficient for the attack, this ICMP advertising an MTU close or below the minimum MTU any link technology must support: 576 in IPv4 and 1280 in IPv6. Because of the IPsec ESP encapsulation, the IPsec gateway then advertises an MTU below this minimum to the local hosts, thereby creating confusion among them. The consequences on the end-hosts can be serious, ranging from performance impacts to Denial of Services (DoS), depending on the exact configuration: operating system (e.g., Linux versus Windows), protocol (e.g., TCP

versus UDP or IPv4 versus IPv6), and internal parameters (e.g., PMTUd versus PLPMTUd).

The present document details both the attack and its consequences on the end-host, depending on the exact configuration. Note that some parts of the present document are not specific to IPsec and similar attacks could be launched in other situations where a gateway need to encapsulate some traffic in a tunnel.

2. Notations, Definitions and Abbreviations

2.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Abbreviations

This document uses the following abbreviations.

ICMP PTB:

> Either an ICMP "Destination unreachable"/"Fragmentation needed" packet (IPv4) or ICMPv6 "Packet Too Big"/"Fragmentation needed" (IPv6), depending on the context

PTB-PTS:

> Packet Too Big-Packet Too Small

3. About Path MTU discovery

Path MTU discovery (PMTUd) is a key mechanism for optimum network performance since it enables a sender to determine the appropriate packet size along a path dynamically (the path may change over time). Two complementary PMTU discovery algorithms are in use: PMTUd and PLPMTUd.

3.1. The legacy PMTUd mechanism

PMTUd [RFC1191] is the legacy approach. Let us illustrate its behavior in an IPv4 (resp. IPv6) network. A sender sets the IPv4 Don’t Fragment (DF) bit in a packet (useless in IPv6 as fragmentation is prohibited). If a router cannot transmit this packet because of its size, it must send back to the sender an ICMP "Destination unreachable"/"Fragmentation needed" packet (resp. an ICMPv6 "Packet Too Big"/"Fragmentation needed"), along with the next hop MTU

information. In the following we will call these error packets ICMP PTB (Packet Too Big), regardless of whether IPv4 or IPv6 is used. Iteratively, upon receiving such an ICMP PTB packet, the sender decreases the packet size until it reaches the lowest MTU on the path to the destination. The PMTU is then found and will be used by the sender for outgoing packets sent to this destination. Since the path can change dynamically (e.g., due to re-routing), this process needs to be performed periodically. Although efficient, the PMTUd approach suffers from several limits, mainly because ICMP packets are often filtered out by some routers/firewalls along their route to the sender. In that case the sender needs another technique to discover the Path MTU.

3.2. The Packetization Layer PMTUd mechanism

To overcome these issues, a new Path MTU discovery mechanism has been developed, that does not rely on ICMP, the Packetization Layer PMTUd (PLPMTUd) [RFC4821]. Instead of using ICMP, it relies on a packetization layer protocol with an acknowledgement mechanism, such as TCP. Using TCP, the sender sends probing packets of a specific size to the destination. If the probing packet is acknowledged, the sender validates that the PMTU is at least equal to the probing packet size, while a time-out indicates that the PMTU is smaller. With TCP, any data segment can be used as a probing packet if enough data is available to fill in the payload. Here also, because the path may change, the PLPMTUd process needs to be performed periodically.

4. The attacker model

In this document, we consider that all the attacks are conducted by adversaries located on the external unsecure black network. We assume an attacker can both eavesdrop the traffic in the IPsec tunnel and inject forged packets. We also assume an attacker has no way to decrypt packets nor encrypt its own packets because the underlying IPsec cryptographic building blocks and key exchange protocols are considered secure. The goal of the attacker is to launch a DoS against the secure tunnel service provided by IPsec gateways, for both kinds of IPsec configurations: host-to-site and site-to-site. Note that in a host-to-site configuration, where a nomad host remotely connects to its home network through an IPsec tunnel, the remote site gateway is the target, not the isolated host.

A requirement is for the attacker to be on the path followed by the IPsec tunnel. For instance, the attacker can be located on a compromised router along the path followed by an IPsec tunnel, in the external unsecure black network. But more simply, the attacker can also be attached to the same unsecure WiFi network (e.g., that sends

WiFi frames in the clear, without any WPA/WPA2 security) as the target user that connects to his home network through an IPsec VPN.

-- Editor's note: the experiments conducted so far are only considering an attacker located on a compromised router along the path. The second configuration, where the attacker takes advantage of an unsecure WiFi network, remains to be tested. --

5. Launching the PTB-PTS attack

5.1. Test configuration

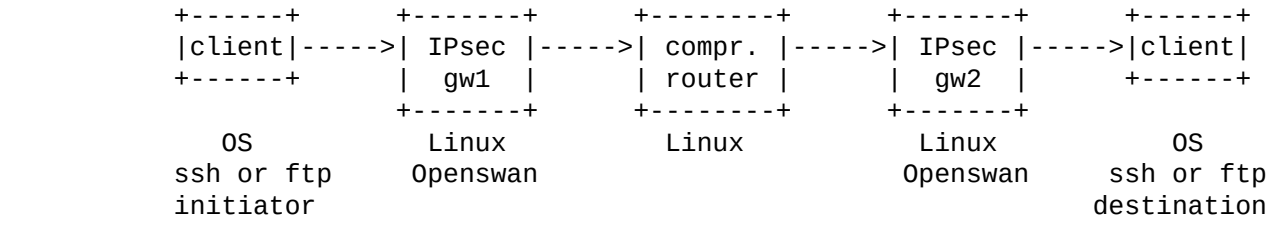

Figure 1: Test configuration.

The results collected in this document have been achieved with the configuration depicted in Figure 1. Five virtual machines are created (with VirtualBox). The compromised router and two IPsec gateways use Ubuntu 14.04.2 LTS. The IPsec gateways use the Openswan IPsec implementation [Openswan] with its default configuration. The two clients use either Ubuntu 14.04.2 LTS or Windows 7. The MTU on the various (virtual) links is configured to the usual 1500 byte Ethernet value.

TCP connection is tested either through ssh (that implies the transmission of keying material whose size is larger than the MTU) or FTP (that implies the transmission of a large file). With both tools, the three-way TCP connection handshake succeeds but problems may arise later when dealing with larger TCP segments.

Windows 7 host configuration is controlled by the EnablePMTUDiscovery [EnablePMTUDiscovery] and EnablePMTUBHDetect [EnablePMTUBHDetect] ("PMTU Black-Hole Detection") registers. The default configuration corresponds to values (1, 0) respectively. We only tested configuration 1-0 (i.e., EnablePMTUDiscovery set to 1 and EnablePMTUBHDetect set to 0).

5.2. Step 1 (common): Forging an ICMP PTB packet from the untrusted network

The attacker first has to forge an appropriate ICMP PTB packet (a single packet is sufficient). This is done by eavesdropping a valid packet from the IPsec tunnel on the untrusted network. Then the attacker forges an ICMP PTB packet, specifying a very small MTU value equal or smaller than 576 with IPv4 (resp. 1280 with IPv6). The attacker can use 0 for instance. This packet spoofs the IP address of a router of the untrusted network (in case the source IP address is checked), and in order to bypass the IPsec protection mechanism against blind attacks, it includes as a payload a part of the outer IP packet that has just been eavesdropped. This is the only packet an attacker needs to send. None of the following steps involve the attacker.

5.3. Step 2 (common): Reset of the PMTU on the gateway

This ICMP packet is the processed by the IPsec gateway. As the packet appears to belong to an active tunnel, the gateway stores the following PMTU value in its SAD: PMTU_SAD = max(MTU_ICMP_PTB, 576) = 576 bytes. It is important to note that the gateway does not store a proposed value smaller than the minimum guaranteed MTU, 576 (resp. 1280) bytes.

At this point, the traffic is not blocked in any way between the targeted gateway and the remote end of the tunnel. Nevertheless the throughput is reduced on the IPsec tunnel as any packet exceeding the PMTU_SAD size must be fragmented (usually by the end-host).

5.4. Following steps with Linux, TCP/IPv4 and PMTUd

The following steps depend on the end-host client Operating System (OS), IP version, and MTU discovery protocol. In case of Linux (Ubuntu 14.04.2 LTS is the OS of all the machines, end-hosts and IPsec gateway), TCP (i.e., during an ssh connection attempt to the remote end-host), IPv4, PMTUd, we observe the following.

The TCP 3-way handshake performs normally as all TCP segments are of tiny size, which enables the attacker to intercept a packet on the IPsec tunnel and to perform the above attack (steps 1 and 2). Then a large packet is sent with the IPv4 Don’t Fragment (DF) bit turned one.

This packet gets rejected by the IPsec gateway as it exceeds the PMTU_SAD value stored in the SAD, and an ICMP PTB error packet is sent back with the following MTU indication: MTU = PMTU_SAD - IP_IPsec_ESP_encapsulation_size. Due to the encapsulation header

(whose size depends on the chosen ciphering algorithm), the gateway restricts the MTU value to 502 bytes.

Upon receiving this ICMP PTB packet, the large TCP segment is fragmented. Nevertheless, instead of creating 502 byte long packets as requested by the gateway, TCP chooses to reduce the MSS to 500 bytes only as it considers the value advertised by the ICMP PTB is below what should be accepted by any link. More precisely, with Linux there is a minimum PMTU configuration parameter (e.g., cat /proc/sys/net/ipv4/route/min_pmtu returns 552 on Debian "Squeeze") that is preferred to the value advertised by the ICMP PTB message: PMTU = max(MTU_ICMP_PTB , PMTU_config). So, once the TCP/IP headers are added, the 500 byte long TCP segment results in a 552 byte long packet.

Since it remains too large, the packet is dropped by the gateway and this latter replies with the same ICMP PTB packets, with the same result.

After 2 minutes of failures and a total of 10 re-transmission attempts, the ssh server closes the connection (FIN/ACK exchange quickly followed by a RST). The DoS successfully prevented any ssh setup.

5.5. Following steps with Linux, TCP/IPv4 and PLPMTUd

In case of TCP/IPv4 and PLPMTUd, we observe the following.

The TCP 3-way handshake performs normally. Then a large packet is sent with the IPv4 Don’t Fragment (DF) bit turned one.

This packet gets rejected by the IPsec gateway and an ICMP PTB is returned to the end-host that restricts the MTU to 502 bytes.

Although PLPMTUd is not dependant on ICMP, this error message is immediately taken into account and several TCP segments of maximum size 500 bytes are sent. Once the TCP/IP headers are added, the 500 byte long TCP segment results in a 552 byte long packet.

Since it remains too large, the packet is dropped by the gateway and this latter replies with the same ICMP PTB packets, with the same result.

This pattern happens 5 times, generating a total of 6 ICMP PTB packets including the first one.

Then, 6.3 seconds after the TCP connection establishment, the PLPMTUd component decides to drastically reduce the segment size: instead of

500 byte TCP segments, it now sends a sequence of alternatively 256 byte TCP segment followed by a 244 byte TCP segment. Those segment are typically probes, chosen by PLPMTUd in order to test this value.

Since the resulting packets are Small enough (at most 256 + 52 = 308 bytes), they reach the other side that acknowledges them. The ssh connection finishes after a few additional segments and a prompt appears in the terminal.

To conclude a delay of 6.3s was required for the ssh connection to be setup. Additionally, any packet leaving the host after this initial delay contains at most 256 bytes of TCP payload, which significantly reduces the TCP throughput and consumes more resources in the forwarding nodes.

5.6. Following steps with Linux, TCP/IPv6 and PMTUd

In case of TCP/IPv6 and PMTUd, we observe the following.

The situation is pretty the same as with IPv4. The main difference is the ICMP PTB packet that advertises a MTU of 1198 bytes (i.e., 1280 minus the various IPsec encapsulation headers). Upon receiving this ICMP PTB packet, the large TCP segment is fragmented into TCP segments of maximum size 1200 bytes. Once the TCP/IPv6 headers are added, it results into packets of maximum size 1256 bytes, which is too large for the IPsec gateway.

After 2 minutes of failures and a total of 10 re-transmission attempts, the ssh server closes the connection (FIN/ACK exchange quickly followed by a RST). The DoS successfully prevented any ssh setup.

5.7. Following steps with Linux, TCP/IPv6 and PLPMTUd

In case of TCP/IPv6 and PLPMTUd, we observe the following.

The situation is pretty the same as with IPv4. However upon receiving the ICMP PTB packet that advertises a MTU of 1198 bytes, the end-host first tries to use TCP MSS=1200 which is too large for the IPsec gateway. A total of 4 re-transmissions happen, generating a total of 5 ICMP PTB packets including the first one.

Then, 3.3 seconds after the beginning, the end-host tries with TCP MSS=504. Once the TCP/IPv6 headers are added, it results into packets of maximum size 560 bytes, which is acceptable for the IPsec gateway. The ssh connection finishes after a few additional segments and a prompt appears in the terminal.

To conclude a delay of 3.3s was required for the ssh connection to be setup (compared to 6.3 in case of IPv4). Additionally, any packet leaving the host after this initial delay contains at most 504 bytes of TCP payload, far below the 1280 minimum MTU guaranteed by IPv6. This behavior significantly reduces the TCP throughput and consumes more resources in the forwarding nodes.

5.8. Following steps with Windows 7, TCP/IPv4 and default PMTU discovery

In case of TCP/IPv4 and PMTU-1-0 (default configuration), we observe the following.

The TCP 3-way handshake performs normally. Then a large packet is sent with the IPv4 Don't Fragment (DF) bit turned one. This packet gets rejected by the IPsec gateway which returns an ICMP PTB with the MTU value to 502 bytes.

Upon receiving this ICMP PTB packet, the Windows 7 end-host sends a smaller TCP segment, of size 556 bytes (instead of 502 bytes). Once the TCP/IP headers are added, this TCP segment results in a 596 byte long packet.

This packet is still too large for the IPsec gateway, however the IPv4 DF bit is now turned off, which authorizes the IPsec gateway to perform IP fragmentation. It therefore gets fragmented by IP within the IPsec gateway, then reassembled on the other side of the tunnel, and acknowledged.

Upon receiving this TCP acknowledgment, the PLPMTUd mechanism starts (Windows 7 merges PMTUd and PLPMTUd like mechanisms together). The following TCP segment is now of size 1112 (i.e., 1152 with TCP/IPv4 headers), here also with the DF bit turned off. This large packet therefore gets fragmented by IP within the IPsec gateway, then reassembled on the other side of the tunnel, and acknowledged by the remote TCP.

The process continues, with TCP segment sizes that progressively increase (we observed a maximum value of 63,940 bytes!), always with the DF bit turned off. All of them are IP fragmented into small IP datagrams of size 548 bytes or 120 bytes. The traffic on the IPsec tunnel is therefore composed of a many tiny packets (never more than 548 bytes long), which creates a huge performance penalty in case of high rate data flows.

5.9. Following steps with Windows 7, TCP/IPv6 and default PMTU discovery

In case of TCP/IPv6 and PMTU-1-0 (default configuration), we observe the following.

The TCP 3-way handshake performs normally. Then a large packet is sent. This packet gets rejected by the IPsec gateway which returns an ICMP PTB with the MTU value set to 1198 bytes (as in Section 5.6).

Upon receiving this ICMP PTB packet, TCP segments have maximum size 1212 bytes (1276 bytes with the TCP/IPv6 headers), which is too large for the IPsec gateway. However the end-host, upon receiving the same ICMP PTB packet, keeps on using MSS=1212, resulting in the same problem.

After 10 transmission attempts and 21 seconds, the ftp client closes the connection (with a RST). The DoS successfully prevented any ssh setup.

5.10. Other configurations

Several tcpdump traces, collected when the end-hosts and gateways run a stable "Squeeze" Debian distribution (instead of Ubuntu), can be found in [Jacquin14]. Note there are some differences (e.g., ICMP PTB proposes an MTU of 514 bytes rather than 502).

6. Summary of the results

The following tables summarize the consequences of a PTB-PTS attack on a host located in the secure network, behind the IPsec gateway.

6.1. Linux end-hosts

Conditions	Results of a PTB-PTS attack
TCP/IPv4, PMTUd	DoS: no ssh connection setup (TCP close after 2 mn)
TCP/IPv4, PLPMTUd	Major performance impacts: initial 6.3 second delay, then tiny packets (TCP MSS=256)
UDP/IPv4, PMTUd	Major performance impacts: tiny packets
TCP/IPv6, PMTUd	DoS: no ssh connection setup (TCP close after 2 mn)
TCP/IPv6, PLPMTUd	Important performance impacts: initial 3.3 second delay, then small packets (TCP MSS=504)
UDP/IPv6, PMTUd	TODO

Results of the attack when the hosts run Linux (Ubuntu 14.04.2 LTS) and IPsec gateways run Linux (Ubuntu 14.04.2 LTS)

6.2. Windows 7 end-hosts

Conditions	Results of a PTB-PTS attack
TCP/IPv4, PMTU-1-0	Functional, but performance impacts (548 and 120 byte IP fragments)
UDP/IPv4	TODO
TCP/IPv6, PMTU-1-0	DoS: no ftp transfer possible (TCP close after 21 sec)
UDP/IPv6, PMTUd	TODO

Results of the attack when the hosts run Windows 7 and IPsec gateways run Linux (Ubuntu 14.04.2 LTS)

7. Discussion

7.1. The two core issues

This work highlights two issues:

Issue 1: determining the legitimacy of untrusted ICMP PTB packets

The two security measures W.R.T. the processing of ICMP/ICMPv6 packets in IPsec and in particular the ICMP PTB packets, namely the outer header verification and payload verification, are

essential to avoid blind attacks, but not sufficient if the attacker is on the path followed by the IPsec tunnel. This is a fundamental limit of the current IPsec specifications.

Issue 2: dealing with minimum Path MTU in presence of a tunnel

When the Path MTU advertised to the IPsec gateway approaches the minimum MTU each link technology should support (i.e., 576 bytes or 1280), problems can arise as IPsec tunnelling adds the IP/IPsec/ESP headers. There are two sides to the problem:

First of all, at the end-host level, we observe that a Linux host does not accept the Path MTU advertised by the IPsec gateway if it is smaller than the minimum MTU configured locally. Indeed, the local component that takes this decision is not aware that the gateway operates an IPsec tunnel and needs some additional room. With the PMTUd approach, the compliance on the minimum MTU is strict and a DoS results. With the PLPMTUd approach, the end-host pragmatically uses (after some time) a TCP segment size significantly lower than this locally configured minimum MTU and the Path MTU advertised by the IPsec gateway. Communications are feasible, but in a sub-optimal way.

The second side of the problem is that the IPsec gateway should not accept from the black network an ICMP PTB asking to reduce the MTU to 576 bytes (resp. 1280 with IPv6). However there is a fundamental contradiction here since 576 bytes (resp. 1280) is a valid MTU value for a link. This is typically a situation where an alarm should be sent to the IPsec gateway administrator, which is not the case today.

7.2. Trivial unsatisfying counter-measures

A trivial counter measure to mitigate the attack consists in configuring the IPsec gateway so that IPv4 packets are fragmented regardless of the original DF bit setting. This is feasible (and recommended) with Cisco IOS 12.2(11)T and above ([Cisco-DF], "DF Bit Override Functionality with IPsec Tunnels" section). However, as mentioned in [Cisco-DF], "a significant performance impact occurs at high data rate", and ignoring the DF bit cannot be considered as a valid approach.

Another trivial counter measure consists in ignoring all ICMP PTB packets coming from the unsecure network. However, this choice compromises PMTUd that the use of PLPMTUd within end-hosts will not totally compensate (e.g., PLPMTUd is only applicable to protocols like TCP relying on acknowledgements, no to UDP).

7.3. Potential solutions

A more interesting choice consists in refusing to reduce the MTU below 576 (resp. 1280) after subtracting the IP/IPsec/ESP encapsulation headers (e.g., an alarm should be sent to the IPsec gateway administrator if this happens). Doing so, the MTU advertised to end-host in ICMP PTB messages will always be at least equal to the minimum MTU any link should provide, which avoids the initial delays and denial of service discussed in this document within the end-hosts. This is not totally satisfying though, given that the attack succeeds in reducing the PMTU.

Therefore, in addition to refusing to reduce the MTU below the minimum MTU, a second complementary approach could be the following: since the legitimacy of an untrusted ICMP packet cannot be determined, the IPsec gateway should to confirm the information with a side mechanism, a PLPMTUd-like probing. It could work as follows. The gateway generates a probing packet of a certain size and that is sent inside the IPsec tunnel. Upon reception, the remote IPsec gateway acknowledges this probe, otherwise a timeout occurs at the gateway that sent the probe. Therefore, if the probing mechanism does not confirm the ICMP PTB information received from the unsecure Internet, this ICMP packet is simply ignored.

For this mechanism to be effective, the attacker should not be able to identify in real time the probing packets in the tunnel in order to discard them selectively. However being able to drop selectively some packets goes far beyond the attacker model considered in this document (Section 4). Additionally, an IPsec gateway level probing mechanism, done periodically, cannot be as reactive as the PMTUd approach (assuming ICMP packets are not filtered out) in case of path MTU change, for instance after a change of route. Therefore we believe that both mechanisms could safely work in parallel.

8. Security Considerations

This I-D is all about security. It identifies a potential attack that leverages on IPsec to attack end-hosts located behind the gateways, in the secure networks. The attack effectiveness depends on the exact end-host configuration: operating system, transport protocol, IP version, MTU discovery mechanism, as detailed in this document.

9. IANA Considerations

N/A

10. Acknowledgments

The authors would like to acknowledge Ludovic Jacquin as the main contributor to the first experiments and author of [Jacquin14].

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[Cisco-DF]
Cisco Systems, , "IPsec Data Plane Configuration Guide, Cisco IOS Release 15MT", 2012, <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dplane/configuration/15-mt/sec-ipsec-data-plane-15-mt-book.pdf>.

[EnablePMTUBHDetect]
Microsoft TechNet, , "EnablePMTUBHDetect", <https://technet.microsoft.com/en-us/library/cc960465.aspx>.

[EnablePMTUDiscovery]
Microsoft TechNet, , "EnablePMTUDiscovery", <https://technet.microsoft.com/en-us/library/cc957539.aspx>.

[Jacquin12]
Jacquin, L., Roca, V., Kaafar, M., Schuler, F., and J-L. Roch, "IBTrack: An ICMP Black holes Tracker", IEEE Global Communications Conference (GLOBECOM'12) , December 2012, <https://hal.inria.fr/hal-00695746/en/>.

[Jacquin14]
Jacquin, L., Roca, V., and J-L. Roch, "Too Big or Too Small? The PTB-PTS ICMP-based Attack against IPsec Gateways", IEEE Global Communications Conference (GLOBECOM'14) , December 2014, <https://hal.inria.fr/hal-01052994/en/>.

[Openswan]
"Openswan", <https://www.openswan.org/>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

Authors' Addresses

Vincent Roca
INRIA
655, av. de l'Europe
Inovallee; Montbonnot
ST ISMIER cedex 38334
France

Email: vincent.roca@inria.fr
URI: http://privatics.inrialpes.fr/people/roca/

Saikou Fall
INRIA
655, av. de l'Europe
Inovallee; Montbonnot
ST ISMIER cedex 38334
France

Email: saikou.fall@inria.fr