Internet Engineering Task Force (IETF)
Request for Comments: 7359
Category: Informational
ISSN: 2070-1721

F. Gont
Huawei Technologies

August 2014

Layer 3 Virtual Private Network (VPN) Tunnel Traffic Leakages in Dual-Stack Hosts/Networks

Abstract

The subtle way in which the IPv6 and IPv4 protocols coexist in typical networks, together with the lack of proper IPv6 support in popular Virtual Private Network (VPN) tunnel products, may inadvertently result in VPN tunnel traffic leakages. That is, traffic meant to be transferred over an encrypted and integrity-protected VPN tunnel may leak out of such a tunnel and be sent in the clear on the local network towards the final destination. This document discusses some scenarios in which such VPN tunnel traffic leakages may occur as a result of employing IPv6-unaware VPN software. Additionally, this document offers possible mitigations for this issue.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7359.

IESG Note

This document describes a problem of information leakage in VPN software and attributes that problem to the software's inability to deal with IPv6. We do not think this is an appropriate characterization of the problem. It is true that when a device supports more than one address family, the inability to apply policy to more than one address family on that device is a defect. Despite that, inadvertent or maliciously induced information leakage may also occur due to the existence of any unencrypted interface allowed on the system, including the configuration of split tunnels in the VPN software itself. While there are some attacks that are unique to an IPv6 interface, the sort of information leakage described by this document is a general problem that is not unique to the situation of IPv6-unaware VPN software. We do not think this document sufficiently describes the general issue.

Table of Contents

1. Introduction

It is a very common practice for users to employ VPN software when employing a public (and possibly rogue) local network. This is typically done not only to gain access to remote resources that may not otherwise be accessible from the public Internet, but also to secure the host's traffic against attackers that might be connected to the same local network as the victim host. The latter case constitutes the problem space of this document. Indeed, it is sometimes assumed that employing a VPN tunnel makes the use of insecure protocols (e.g., that transfer sensitive information in the clear) acceptable, as a VPN tunnel provides security services (such as data integrity and/or confidentiality) for all communications made over it. However, this document illustrates that under certain circumstances, some traffic might not be mapped onto the VPN tunnel and thus might be sent in the clear on the local network.

Many VPN products that are typically employed for the aforementioned VPN tunnels only support the IPv4 protocol: that is, they perform the necessary actions such that IPv4 traffic is sent over the VPN tunnel, but they do nothing to secure IPv6 traffic originated from (or being received at) the host employing the VPN client. However, the hosts themselves are typically dual-stacked: they support (and enable by default) both IPv4 and IPv6 (even if such IPv6 connectivity is simply "dormant" when they connect to IPv4-only networks). When the IPv6 connectivity of such hosts is enabled, they may end up employing an IPv6-unaware VPN client in a dual-stack network. This may have "unexpected" consequences, as explained below.

The subtle way in which the IPv4 and IPv6 protocols interact and coexist in dual-stacked networks might, either inadvertently or as a result of a deliberate attack, result in VPN tunnel traffic leakages -- that is, traffic meant to be transferred over a VPN tunnel could leak out of the VPN tunnel and be transmitted in the clear from the local network to the final destination, without employing the VPN services at all.

Since this issue is specific to VPN solutions that route Layer 3 traffic, it is applicable to the following types of VPN technologies:

- IPsec-based VPN tunnels

- SSL/TLS VPN tunnels

  NOTE: see Section 2 for a definition and description of these terms.

Section 2 clarifies the terminology employed throughout this document. Section 3 provides some background about IPv6 and IPv4 coexistence, summarizing how IPv6 and IPv4 interact on a typical dual-stacked network. Section 4 describes the underlying problem that leads to the aforementioned VPN traffic leakages. Section 5 describes legitimate scenarios in which such traffic leakages might occur, while Section 6 describes how VPN traffic leakages can be triggered by deliberate attacks. Finally, Section 7 discusses possible mitigations for the aforementioned issue.

2. Terminology

When employing the term "Virtual Private Network tunnel" (or "VPN tunnel"), this document refers to VPN tunnels based on IPsec or SSL/TLS, where Layer 3 packets are encapsulated and sent from a client to a middlebox, to access multiple network services (possibly employing different transport and/or application protocols).

IPsec-based VPN tunnels simply employ IPsec in tunnel mode to encapsulate and transfer Layer 3 packets over the VPN tunnel. On the other hand, the term "SSL/TLS-based VPN tunnels" warrants a clarification, since two different technologies are usually referred to as "SSL/TLS VPN":

SSL/TLS VPN tunnel:
: A technology that encapsulates traffic from a client to a middlebox. In essence, an SSL/TLS VPN tunnel acts just like an IPsec-based tunnel, but instead employs SSL/TLS for encryption and some proprietary/unspecified mechanism for encapsulation and routing.

SSL/TLS VPN portal:
A front-end provided by the middlebox to add security to a normally unsecured site. An SSL/TLS VPN portal is typically application specific, wrapping the specific protocol, such as HTTP, to provide access to specific services on a network. In such a case, the SSL/TLS VPN portal would be accessed just like any HTTPS URL. SSL/TLS VPN portals are used when one wants to restrict access and only provide remote users to very specific services on the network. SSL/TLS VPN portals do not require an agent, and the policy is typically more liberal from organizations allowing access from anywhere via this access method. All other traffic on the system may be routed directly to the destination, whether it is IPv4, IPv6, or even other service level (HTTP) destination addresses.

Our document focuses on Layer 3 VPNs that employ IPsec-based or SSL/TLS-based tunnels, and excludes the so-called SSL/TLS VPN portals. Both IPsec-based and SSL/TLS-based VPN tunnels are designed to route Layer 3 traffic via the VPN tunnel through to the VPN tunnel server. Typically, these solutions are agent based, meaning that software is required on the client endpoint to establish the VPN tunnel and manage access control or routing rules. This provides an opportunity for controls to be managed through that application as well as on the client endpoint.

NOTE: Further discussion of SSL-based VPNs can be found in [SSL-VPNs].

We note that, in addition to the general case of "send all traffic through the VPN", this document considers the so-called "split-tunnel" case, where some subset of the traffic is sent through the VPN, while other traffic is sent to its intended destination with a direct routing path (i.e., without employing the VPN tunnel). We note that many organizations will prevent split-tunneling in their VPN configurations if they would like to make sure the users data goes through a gateway with protections (malware detection, URL filtering, etc.), but others are more interested in performance of the user’s access or the ability for researchers to have options to access sites they may not be able to through the gateway.

3. IPv4 and IPv6 Coexistence

The coexistence of the IPv4 and IPv6 protocols has a number of interesting and subtle aspects that may have "surprising" consequences. While IPv6 is not backwards-compatible with IPv4, the two protocols are "glued" together by the Domain Name System (DNS).

For example, consider a site (say, www.example.com) that has both IPv4 and IPv6 support. The corresponding domain name (www.example.com, in our case) will contain both A and AAAA DNS resource records (RRs). Each A record will contain one IPv4 address, while each AAAA record will contain one IPv6 address -- and there might be more than one instance of each of these record types. Thus, when a dual-stacked client application means to communicate with www.example.com, it can request both A and AAAA records and use any of the available addresses. The preferred address family (IPv4 or IPv6) and the specific address that will be used (assuming more than one address of each family is available) varies from one protocol implementation to another, with many host implementations preferring IPv6 addresses over IPv4 addresses.

> NOTE: [RFC6724] specifies an algorithm for selecting a destination address from a list of IPv6 and IPv4 addresses. [RFC6555] discusses the challenge of selecting the most appropriate destination address, along with a proposed implementation approach that mitigates connection-establishment delays.

As a result of this "coexistence" between IPv6 and IPv4, when a dual-stacked client means to communicate with some other system, the availability of A and AAAA DNS resource records will typically affect which protocol is employed to communicate with that system.

4. Virtual Private Networks in IPv4/IPv6 Dual-Stack Hosts/Networks

Many VPN tunnel implementations do not support the IPv6 protocol -- or, what is worse, they completely ignore IPv6. This typically means that, once a VPN tunnel has been established, the VPN software takes care of the IPv4 connectivity by, e.g., inserting an IPv4 default route that causes all IPv4 traffic to be sent over the VPN tunnel (as opposed to sending the traffic in the clear, employing the local router). However, if IPv6 is not supported (or completely ignored), any packets destined to an IPv6 address will be sent in the clear using the local IPv6 router. That is, the VPN software will do nothing about the IPv6 traffic.

The underlying reason for which these VPN leakages may occur is that, for dual-stacked systems, it is not possible to secure the communication with another system without securing both protocols (IPv6 and IPv4). Therefore, as long as the traffic for one of these protocols is not secured, there is the potential for VPN traffic leakages.

5. Inadvertent VPN Tunnel Traffic Leakages in Legitimate Scenarios

Consider a dual-stacked host that employs IPv4-only VPN software to establish a VPN tunnel with a VPN server, and that said host now connects to a dual-stacked network (that provides both IPv6 and IPv4 connectivity). If some application on the client means to communicate with a dual-stacked destination, the client will typically query both A and AAAA DNS resource records. Since the host will have both IPv4 and IPv6 connectivity, and the intended destination will have both A and AAAA DNS resource records, one of the possible outcomes is that the host will employ IPv6 to communicate with the intended destination. Since the VPN software does not support IPv6, the IPv6 traffic will not employ the VPN tunnel; hence, it will have neither integrity nor confidentiality protection from the source host to the final destination.

This could inadvertently expose sensitive traffic that was assumed to be secured by the VPN software. In this particular scenario, the resulting VPN tunnel traffic leakage is a side effect of employing IPv6-unaware VPN software in a dual-stacked host/network.

6. VPN Tunnel Traffic Leakage Attacks

A local attacker could deliberately trigger IPv6 connectivity on the victim host by sending forged ICMPv6 Router Advertisement messages [RFC4861]. Such packets could be sent by employing standard software such as rtadvd [RTADVD], or by employing packet-crafting tools such as SI6 Network’s IPv6 Toolkit [SI6-Toolkit] or THC’s IPv6 Attack Toolkit [THC-IPv6]. Once IPv6 connectivity has been enabled, communications with dual-stacked systems could result in VPN tunnel traffic leakages, as previously described.

While this attack may be useful enough (due to the increasing number of IPv6-enabled sites), it will only lead to traffic leakages when the destination system is dual-stacked. However, it is usually trivial for an attacker to trigger such VPN tunnel traffic leakages for any destination system: an attacker could simply advertise himself as the local recursive DNS server by sending forged Router Advertisement messages [RFC4861] that include the corresponding Recursive DNS Server (RDNSS) option [RFC6106], and then perform a DNS spoofing attack such that he can become a "man in the middle" and intercept the corresponding traffic. As with the previous attack scenario, packet-crafting tools such as [SI6-Toolkit] and [THC-IPv6] can readily perform this attack.

NOTE: Some systems are known to prefer IPv6-based recursive DNS servers over IPv4-based ones; hence, the "malicious" recursive DNS servers would be preferred over the legitimate ones advertised by the VPN server.

7. Mitigations to VPN Tunnel Traffic Leakage Vulnerabilities

At the time of this writing, a large number of VPN implementations have not yet addressed the issue of VPN tunnel traffic leakages. Most of these implementations simply ignore IPv6; hence, IPv6 traffic leaks out of the VPN tunnel. Some VPN tunnel implementations handle IPv6, but not properly. For example, they may be able to prevent inadvertent VPN tunnel traffic leakages arising in legitimate dual-stack networks, but they may fail to properly handle the myriad of vectors available to an attacker for injecting "more specific routes", such as ICMPv6 Router Advertisement messages with Prefix Information Options and/or Route Information Options, and ICMPv6 Redirect messages.

Clearly, the issue of VPN tunnel traffic leakages warrants proper IPv6 support in VPN tunnel implementations.

7.1. Fixing VPN Client Software

There are a number of possible mitigations for the VPN tunnel traffic leakage vulnerability discussed in this document.

If the VPN client is configured by administrative decision to redirect all IPv4 traffic to the VPN, it should:

1. If IPv6 is not supported in the VPN software, disable IPv6 support in all network interfaces.

   NOTE: For IPv6-unaware VPN clients, the most simple mitigation would be to disable IPv6 support in all network interface cards when a VPN tunnel is meant to be employed. Thus, applications on the host running the VPN client software will have no other option than to employ IPv4; hence, they will simply not even try to send/process IPv6 traffic. We note that this should only be regarded as a temporary workaround, since the proper mitigation would be to correctly handle IPv6 traffic.

2. If IPv6 is supported in the VPN software, ensure that all IPv6 traffic is also sent via the VPN.

   NOTE: This would imply, among other things, properly handling any vectors that might be employed by an attacker to install IPv6 routes at the victim system (such as ICMPv6 Router Advertisement messages with Prefix Information Options or Route information Options [RFC4191], ICMPv6 Redirect messages, etc.). We note that properly handling all the aforementioned vectors may prove to be nontrivial.

If the VPN client is configured to only send a subset of IPv4 traffic to the VPN tunnel (split-tunnel mode), then:

1. If the VPN client does not support IPv6, it should disable IPv6 support in all network interfaces.

   NOTE: As noted above, this should only be regarded as a temporary workaround, since the proper mitigation would be to correctly handle IPv6 traffic.

2. If the VPN client supports IPv6, it is the administrators responsibility to ensure that the correct corresponding sets of IPv4 and IPv6 networks get routed into the VPN tunnel.

   NOTE: As noted above, this would imply, among other things, properly handling any vectors that might be employed by an attacker to install IPv6 routes at the victim system. This may prove to be a nontrivial task.

A network may prevent local attackers from successfully performing the aforementioned attacks against other local hosts by implementing First-Hop Security solutions such as Router Advertisement Guard (RA-Guard) [RFC6105] and DHCPv6-Shield [DHCPv6-SHIELD]. However, for obvious reasons, a host cannot and should not rely on this type of mitigations when connecting to an open network (cybercafe, etc.).

NOTE: Besides, popular implementations of RA-Guard are known to be vulnerable to evasion attacks [RFC7113].

Finally, we note that if (eventually) IPv6-only VPN implementations become available, similar issues to the ones discussed in this document could arise if these IPv6-only VPN implementations do nothing about the IPv4 traffic.

7.2. Operational Mitigations

While the desired mitigation for the issues discussed in this document is for VPN clients to be IPv6 aware, we note that in scenarios where this would be unfeasible, an administrator may want to disable IPv6 connectivity on all network interfaces of the node employing the IPv6-unaware VPN client.

8. Security Considerations

This document discusses how traffic meant to be transferred over a VPN tunnel can leak out of such a tunnel and, hence, appear in the clear on the local network. This is the result of employing IPv6-unaware VPN client software on dual-stacked hosts.

The proper mitigation of this issue is to correctly handle IPv6 traffic in the VPN client software. However, in scenarios in which such a mitigation is unfeasible, an administrator may choose to disable IPv6 connectivity on all network interfaces of the host employing the VPN client.

9. Acknowledgements

The author would like to thank Kathleen Moriarty and Paul Hoffman who contributed text that was readily incorporated into Section 2 of this document.

The author of this document would like to thank (in alphabetical order) Cameron Byrne, Spencer Dawkins, Gert Doering, Stephen Farrell, Seth Hall, Paul Hoffman, Tor Houghton, Russ Housley, Joel Jaeggli, Alastair Johnson, Merike Kaeo, Panos Kampanakis, Stephen Kent, Henrik Lund Kramshoj, Warren Kumari, Barry Leiba, Kathleen Moriarty, Thomas Osterried, Jim Small, Martin Vigoureux, and Andrew Yourtchenko for providing valuable comments on earlier draft versions of this document.

The author wishes to express deep and heartfelt gratitude to Enrique Garcia and Vicenta Tejedo, for their precious love and support.

10. References

10.1. Normative References

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

10.2. Informative References

[DHCPv6-SHIELD] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers", Work in Progress, July 2014.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.

[RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, February 2014.

[RTADVD] "rtadvd(8) manual page", <http://www.freebsd.org/cgi/man.cgi?query=rtadvd&sektion=8>.

[SI6-Toolkit] SI6 Networks, "SI6 Networks' IPv6 Toolkit", <http://www.si6networks.com/tools/ipv6toolkit>.

[SSL-VPNs] Hoffman, P., "SSL VPNs: An IETF Perspective", IETF 72, SAAG Meeting, 2008, <http://www.ietf.org/proceedings/72/slides/saag-4.pdf>.

[THC-IPv6] The Hacker's Choice, "THC-IPV6 - attacking the IPV6 protocol suite", <http://www.thc.org/thc-ipv6/>.

Author's Address

Fernando Gont
Huawei Technologies
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
EMail: fgont@si6networks.com
URI: http://www.si6networks.com