Monitoring of Tunneled IPv6 Traffic Using Packet Decapsulation and IPFIX

Martin Elich[1,3], Matěj Grégr[1,2] and Pavel Čeleda[1,3]

[1] CESNET, z.s.p.o., Prague, Czech Republic
[2] Brno University of Technology, Brno, Czech Republic – igregr@fit.vutbr.cz
[3] Masaryk University, Brno, Czech Republic – {elich|celeda}@mail.muni.cz

Abstract. IPv6 is being deployed but many Internet Service Providers have not implemented its support yet. Most of the end users have IPv6 ready computers but their network doesn't support native IPv6 connection so they are forced to use transition mechanisms to transport IPv6 packets through IPv4 network. We do not know, what kind of traffic is inside of these tunnels, which services are used and if the traffic does not bypass security policy. This paper proposes an approach, how to monitor IPv6 tunnels even on high-speed networks. The proposed approach allows to monitor traffic on 10 Gbps links, because it supports hardware-accelerated packet distribution on multi-core processors. A system based on the proposed approach is deployed at the CESNET2 network, which is the largest academic network in the Czech Republic. This paper also presents several statistics about tunneled traffic on the CESNET2 backbone links.

Keywords: IPv6, Teredo, ISATAP, 6to4, network monitoring, IPv6 tunnel, IPFIX, FlowMon

1 Introduction

End users have nowadays IPv6 ready computers, because support for this protocol is available in main operating systems (Windows, Linux, BSD, Mac OS X). Unfortunately, not every ISP has implemented IPv6 support yet, which together with IPv6 backward incompatibility with IPv4 protocol requires transition mechanisms. 6to4, Teredo and ISATAP are the most used transition techniques. These three methods use encapsulation of IPv6 protocol inside IPv4 protocol – tunneling. The encapsulation hides the IPv6 traffic. Tunneled traffic may look like ordinary IPv4 traffic using UDP ports, so administrators do not know, which IPv6 network service is requested, how much traffic flows through tunnels etc. IPv6 tunnels are created automatically so there is no need for a user intervention. This can cause security problems such as bypassing firewalls, unauthorized use of services etc.

We propose an approach how to overcome this limitation and how to monitor tunneled IPv6 traffic. It features hardware-accelerated packet distribution with which it is possible to monitor even 10 Gbps links. Statistics and tunneled traffic distribution presented in this paper are generated from IPFIX data collected on CESNET2 backbone links, which is the largest academic network in the Czech Republic.

The paper is organized as follows. Section 2 describes related work. IPv6 transition techniques are described in Section 3. Proposal of architecture for monitoring tunneled data is in Section 4. Section 5 shows several statistics and analysis from network monitoring and Conclusion is in Section 6.

2 State-of-the-Art and Contribution

Several papers discuss and present IPv6 address and traffic analysis. Authors in [4] analyze traffic from a US Tier-1 ISP. Analyzed traffic in their data-set consists mainly of DNS and ICMP packets. They believe that it is because ISP's customers consider IPv6 traffic still as experimental. For IPv6 address assignment they used methodology introduced in [5]. Statistics from a China Tier-1 ISP are presented in [3]. Their observation about address assignment and application usage are similar to ours with some exceptions. Their traffic contains higher proportion of native IPv6 traffic. We believe, that it is due to larger expansion of IPv6 in China and Asia.

Unfortunately analysis of tunneled IPv6 traffic is missing in many papers. Some statistics are presented in [4] but just for Teredo traffic. Paper [6] observes IPv6 traffic on 6to4 relay but it is quite old. Despite our best efforts we did not find publications about tunneled IPv6 traffic in ISATAP tunnels. Statistics about 6to4 tunnels or Teredo are not so detailed and up to date. This paper tries to update knowledge about nowadays native and tunneled IPv6 traffic.

Contribution of this paper consists of several parts. First, we propose an approach, how to extend IPFIX to provide possibility to monitor tunneled IPv6 traffic. This approach is scalable and can be used in very large networks for monitoring IPv4, native IPv6 and tunneled IPv6 traffic. It is possible to use our concept to collect traffic on high-speed 10 Gbps links with no need to use packet sampling. Second, we present several statistics for tunneling mechanisms. Deployment of IPv6 protocol accelerates because new operating systems use this protocol by default. Therefore more services are accessible through IPv6 protocol and traffic distribution is nowadays completely different than before. Hence current statistics are very useful.

3 Transition Techniques

IPv6 connectivity is enabled and preferred in most operating systems by default. If a station is connected to local IPv4 network without native IPv6 connectivity and web site or another network service is accessible through both protocols, IPv6 has precedence and a host tries to communicate through this protocol first. Because IPv6 is not compatible with the previous IPv4 protocol, different types of transition techniques were proposed. The most interesting are tunneling techniques, because we do not know, which protocols and services are used inside the tunnels. 6to4, Teredo and ISATAP are todays most used tunneling mechanisms for connection to IPv6 network.

6to4 tunneling is the most used transition technique today. According to priority in operating system, if a network device has public IPv4 address, 6to4 is the first mechanism to be used. A host construct 64 bits long IPv6 network prefix according to rules described in [8]. Last 64 bits are used as EUI (End Unit Identifier). Several techniques can be used to create the identifier: based on EUI-64, manual assignment or randomly generated [2]. Default configuration in Windows or Linux use well-known EUI values in practice. Linux use the value 1 by default and Windows XP, Vista, 7 use IPv4 address in lower 32 bits of the EUI [12]. When sending packets, the 6to4 tunnel wraps an IPv6 datagram into an IPv4 datagram with protocol number 41.

Teredo was designed to be able to send network traffic through NAT [7]. It does not encapsulate IPv6 packet in protocol 41 but send it via UDP packet on default port 3544. Teredo address is more complicated then 6to4 and consists of Teredo prefix, Teredo server address, flags, port and client's external address. When simple encapsulation is used only the IPv6 packet is carried as the payload of an UDP packet. Server may insert other fields such as Origin and Authentication.

ISATAP – Intra-Site Automatic Tunnel Addressing Protocol is an IPv6 transition mechanism used in local networks to connect islands of IPv6 nodes over IPv4 networks. Connection to the Internet is made by another mechanism such as 6to4. ISATAP like 6to4 uses encapsulation in protocol number 41 [9]. Nowadays ISATAP is usually the last used transition techniques. Transition techniques order which a host tries when does not have native IPv6 connectivity is usually 6to4, Teredo, ISATAP.

4 Architecture and Implementation

The proposed approach for tunneled IPv6 traffic monitoring describes whole process of IP flow generation, export and collection. The flows are generated by FlowMon exporter a software probe which is able to export NetFlow and IPFIX data. The FlowMon exporter is able to generate flow statistics from any source if the input plug-in supports it [1].

4.1 Architecture

The proposed approach consist of three layers (see Figure 1). The first layer can be a network card or a more specialized hardware. Purpose of this layer is capturing packets and sending them over the software interface to the input plug-in. We used the FPGA based COMBOv2 card and libsze2 library as a software interface. We developed FPGA design for COMBOv2 cards HANIC (Hardware-Accelerated Network Interface Card) which provides a high precision timestamp generated for each packet. Packets can be distributed to several DMA (Direct Memory Access) channels. Packet distribution is one of benefits of proposed approach and is described in Section 4.2.

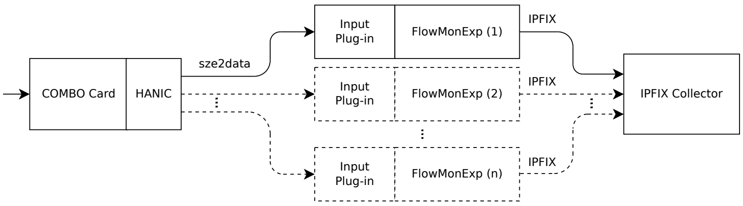

Fig. 1. System architecture – packets are captured by the COMBOv2 card and can be distributed to 16 FlowMon exporters with loaded input plug-in. IP flows are generated based on processed packets and later exported in IPFIX format.

The second layer reads packets from the software interface and processes them with the FlowMon exporter [1]. We designed and implemented input plug-in for monitoring of IPv6 tunneled traffic but plug-ins can have any other functionality.

The plug-in for tunneled IPv6 traffic monitoring detects packets, which are part of tunnels, using a defined set of rules. After tunnel is detected, IPv4 header is stripped out and packets are processed by IPv6 header parser. Relevant information from packet are stored to a data structure representing a part of flow (in this case flow containing single packet). This filled data structure is passed to the exporter. More about plug-in functionality can be found in Section 4.3. The exporter generates flow statistics based on data structures from the input plug-in. Flow statistics are exported in IPFIX using custom IPFIX templates with enterprise-specific information elements to carry information about the tunnel.

The third and last layer is the IPFIX collector.

4.2 Packet Distribution

Packet distribution is implemented using the HANIC design. The goal is to distribute packets between several instances of the FlowMon exporter on the hardware level.

The HANIC design provides a packet header parser. The parser can extract necessary fields for flow identification. The output of parsing unit is a sequence of bits with fixed length of 301 bits. This sequence is then passed to the HASH unit which computes CRC hash with length of $log_2(number\ of\ channels)$. Each packet is send to one of channels according to its hash (the hash is used to address a channel).

Current version of the design use hash length of four bits. This allows to distribute packets to 16 instances of the FlowMon exporter without breaking the flow cache. Another advantage is possibility to process packets on multiple processors which greatly improves overall performance.

4.3 Plug-in Implementation

The input plug-in is implemented as shared library for Linux. It filters and preprocess each packet to data structure compatible with the FlowMon exporter plug-in API. The input plug-in reads packets from the COMBOv2 card in a form of memory chunks. These memory chunks consist of whole packet together with high precision timestamp and card's interface identifier from which packet was read. Protocol number is extracted from Ethernet header or from the MPLS label if MPLS is used.

All IPv4 packets are processed by the filters to detect presence of tunneling. Detection supports the following tunneling mechanisms: Teredo, 6to4 and ISATAP. If Teredo encapsulation is found and encapsulated IPv6 address is in format which is specified in [7], plug-in sets type of tunnel to indicate usage of Teredo and pass filled data structure to exporter. Detection of ISATAP and 6to4 packets is similar as they share some characteristics. IPv4 protocol must be set to value 41. In both mechanisms IPv4 header is followed by IPv6 header. To decide if IPv6 packet is encapsulated by ISATAP or 6to4 plug-in checks IPv6 addresses and looks for address in format specified for 6to4 or ISATAP. Filled data structure is passed to the FlowMon exporter.

4.4 Packet Processing Performance

Packet processing performance was measured as a throughput test when processing packets from 10 Gbps Ethernet link (see Figure 2). The measurement run on 2.0 GHz quad-core CPU and beside throughput we also monitored CPU usage (see Figure 3). Throughput was measured for Teredo and 6to4 packets (throughput of ISATAP packets is the same as throughput of 6to4 packets). In first scenario all packets were processed by single instance of the FlowMon exporter with loaded input plug-in. In the second scenario packets were distributed to 4 instances of the FlowMon exporter with loaded input plug-in. Each instance of the FlowMon exporter was running on different CPU core providing more computing power for processing.

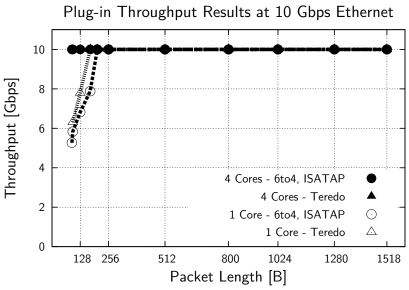

Fig. 2. Throughput on 1 and 4 CPU cores.

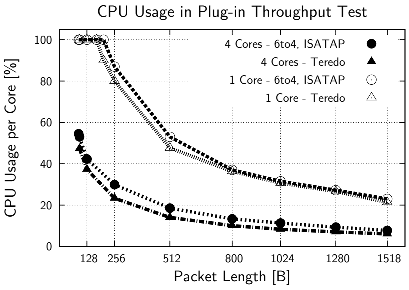

Fig. 3. CPU load on 1 and 4 cores.

To minimize impact of flow generation on performance results all packets in the first scenario originated from single flow. In case of the second scenario four different flows were used.

5 Monitoring of Real Network

We deployed monitoring system based on the proposed approach on the CESNET2 network. Three 10 Gbps backbone links which are connecting the CESNET2 network to SANET (Slovak academic network), PIONIER (Polish optical Internet) and NIX.CZ (Neutral Internet eXchange of not only Czech Republic) networks were monitored. We were forced to slightly change the IPFIX templates in way they shouldn't be according to the IPFIX standard as we were using modified NfSen. NfSen doesn't have full support for enterprise-specific elements [13]. The presented statistics are from September 24 to October 6, 2010.

5.1 IPv6 Address Assignment

Address assignment is a little bit different in IPv6 networks. Usually stateless auto-configuration is used [11], so a host learns just network prefix and default gateway. The lower part of IPv6 address (last 64 bits) is a host identifier and can be assigned manually, based on EUI-64 algorithm or generated randomly according [2]. We use similar methodology for address classification as in [5] but some addresses are analyzed in detail.

Table 1 shows average number of unique IPv6 addresses in native, 6to4, Teredo and ISATAP traffic per day. Surprisingly there is very high number of Teredo addresses. Further examinations showed that Teredo is used mainly for p2p sharing. We believe that it is because BitTorrent clients such as $\mu Torrent$ have implemented Teredo support, to be able to share data with more peers. We detected several Teredo servers as well.

Native and 6to4 addresses are more analyzed and results are shown in Table 2 and Table 3. First table describes in detail 6to4 addresses in native and tunneled traffic. Autoconf means, that EUI is generated according to EUI-64. Linux and Windows rows describes, how many hosts use Windows and Linux/Unix operating systems. This detection is based on default values for the EUI fields [12]. Privacy means, that EUI is generated according to Privacy Extensions. The second table shows address structure of global IPv6 address in native and tunneled traffic.

5.2 Tunneled Traffic Characteristics

The first interesting fact about IPv6 tunneled traffic is, according to our measurement, that it generates more traffic then native IPv6 traffic. This fact is true for all of three metrics (by flow, by packets and by bytes) and is shown in Table 4. As described earlier, the reason for this can be presence of tunneling mechanisms in recent versions of MS Windows.

Traffic	Unique Addresses	Note
Native IPv6	8059 (10.1%)	details in Table 3
6to4	20090 (25.3%)	details in Table 2
Teredo	51330 (64.5%)	detected 13 Teredo servers
ISATAP	82 (0.1%)	

Table 1. IPv6 unique addresses – average per day.

	Native	Tunneled Traffic
Autoconf	2.7%	1.4%
Linux	1.2%	0.3%
Windows	91.2%	85.6%
Privacy	4.9%	12.7%

Table 2. 6to4 addresses in detail.

	Native	Tunneled Traffic
Autoconf	9%	4.2%
Privacy	69.2%	69%
Low	21.8%	26.8%

Table 3. Global IPv6 addresses in detail.

Majority of IPv6 tunneled traffic uses Teredo mechanism (see Table 5). The least used mechanism is ISATAP that may be given by the fact that it is the least preferred option of tunneling in MS Windows.

	Flows	Packets	Bytes
IPv4	98.39%	99.19%	99.13%
Native IPv6	0.10%	0.12%	0.21%
Tunneled IPv6	1.50%	0.69%	0.66%

Table 4. Traffic distribution.

	Flows	Packets	Bytes
Teredo	88.18%	89.10%	88.85%
ISATAP	0.06%	0.03%	0.03%
6to4	11.76%	11.76%	11.12%

Table 5. Tunnel distribution.

We also observed very different distribution of application protocols in tunneled IPv6 traffic. The most used protocol in IPv4 and IPv6 traffic is HTTP. In tunneled IPv6 traffic its share was very small and the traffic was overall spread to hundreds of UDP and TCP ports with high numbers. We come to conclusion that tunneled IPv6 especially Teredo is used for p2p sharing. Reasons, why p2p programs use Teredo are described in Section 5.1.

	Flows			Packets			Bytes		
	IPv4	IPv6	Tunnel	IPv4	IPv6	Tunnel	IPv4	IPv6	Tunnel
HTTP	38.25%	1.99%	0.35%	49.99%	65.50%	2.98%	56.80%	76.16%	0.38%
HTTPS	3.26%	<0.01%	0.08%	1.72%	<0.01%	2.85%	1.17%	<0.01%	0.33%
DNS	10.39%	61.76%	0.45%	0.45%	1.68%	0.05%	0.07%	0.42%	0.01%

Table 6. Protocol distribution in tunneled and native traffic.

6 Conclusion

Current flow-based traffic monitoring techniques can not easily analyze tunneled traffic. It is especially problem in IPv6 networks. In IPv6 networks tunnels are created automatically, without users or administrators intervention. Because IPv6 protocol is not compatible with current IPv4, these tunneling mechanisms would be needed for several years. Network administrators will need an approach, which is able to monitor tunneled traffic on high-speed networks, is scalable and can be integrated into current monitoring systems. In this paper we propose such an approach.

Monitoring 10 Gbps links is possible using hardware-accelerated network cards. We implemented a plug-in for the FlowMon exporter, which can monitor tunneled IPv6 traffic and export obtained data using IPFIX. Collected data can be further analyzed by IDS (Intrusion Detection System) and IPS (Intrusion Prevention System). Current monitoring software miss information about the tunneled traffic. We propose an approach which is able to monitor this kind of traffic. We successfully deployed the proposed solution on academic backbone links in the Czech Republic.

Acknowledgments. This work was supported by the Research Intent of the Czech Ministry of Education MSM6383917201 and by the Czech Ministry of Defence – contract no. SMO02008PR980-OVMASUN200801.

References

1. INVEA-TECH a.s., *FlowMon exporter*, [online], cited [30.09.2010] `http://www.invea-tech.com/products-and-services/flowmon/flowmon-probes`.
2. Narten, T., Draves R., Krishnan S., *RFC 4941, Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September, 2007.
3. Shen W., Chen Y., Zhang Q., et al., *Observations of IPv6 traffic*, In CCCM, 2009, vol. 2, ISBN 978-1-4244-4247-8, p. 278 - 282.
4. Karpilovsky E., Gerber A., Pei D., Rexford J., Shaikh A., *Quantifying the Extent of IPv6 Deployment*, In PAM, 2009, p. 13 - 22.
5. Malone D., *Observation of IPv6 Addresses*, In PAM, 2008, p. 21 - 30.
6. Savola P., *Observations of IPv6 Traffic on a 6to4 Relay*, ACM SIGCOMM CCR vol. 35, no. 1, pp. 23-28, Jan. 2005.
7. Huitema C. *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). RFC 4380*, February 2006.
8. Carpenter B., Moore K. *Connection of IPv6 Domains via IPv4 Clouds. RFC 3056.*
9. Templin D. T. F., Gleeson T., *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). RFC 5214*, March 2008.
10. Nordmark E., Gilligan R., *Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC4213*, October 2005.
11. Thomson S., Narten T., Jinmei T., *IPv6 Stateless Address Autoconfiguration. RFC4862*, September 2007.
12. Warfield H. M., *Security Implication of IPv6, Internet Security Systems*, 2003.
13. Krejčí R., *Network Traffic Collection with IPFIX Protocol*, [online], cited [2010-10-04] `http://is.muni.cz/th/98863/fi_m/xkrejc14_dp.pdf` .