Flow-Based Detection of IPv6-specific Network Layer Attacks

Luuk Hendriks[1(✉)], Petr Velan[2], Ricardo de O. Schmidt[1], Pieter-Tjerk de Boer[1], and Aiko Pras[1]

[1] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands
{luuk.hendriks,r.schmidt,p.t.deboer,a.pras}@utwente.nl
[2] CESNET, a.l.e, Zikova 4, 160 00 Prague 6, Czech Republic
petr.velan@cesnet.cz

Abstract. With a vastly different header format, IPv6 introduces new vulnerabilities not possible in IPv4, potentially requiring new detection algorithms. While many attacks specific to IPv6 have proven to be possible and are described in the literature, no detection solutions for these attacks have been proposed. In this study we identify and characterise IPv6-specific attacks that can be detected using flow monitoring. By constructing flow-based signatures, detection can be performed using available technologies such as NetFlow and IPFIX. To validate our approach, we implemented these signatures in a prototype, monitoring two production networks and injecting attacks into the production traffic.

1 Introduction

Monitoring network traffic is an essential aspect in today's Internet. With the ever-growing collection of possible network-based threats, security officers need to stay up to date and be aware of what is possibly coming towards their networks and services. Intrusion Detection Systems (IDSs) play a critical role in this scenario, offering the first insight into malicious traffic, *e.g.* brute-force attacks on SSH daemons [2], or large numbers of DNS responses caused by a Distributed Denial of Service (DDoS) attack. Currently, the adoption and deployment of IPv6 in the Internet is increasing: 16.4% of users of Google's services have IPv6 connectivity. North-America and Germany feature an adoption of around 30%, and Belgium is almost at 50%. With the increasing amount of IPv6 traffic in mind, we want to know whether the flow-based detection approaches from IPv4 are applicable, and moreover, fully covering the spectrum of IPv6 attacks.

In this paper, we ask ourselves 1. which new threats are introduced by these changes in the network layer; 2. how fundamental these threats are; and 3. how flow-based monitoring solutions should be adapted in order to enable detection of these new attacks.

D. Tuncer et al. (Eds.): AIMS 2017, LNCS 10356, pp. 137–142, 2017.
DOI: 10.1007/978-3-319-60774-0_11

2 Methodology

We focus on a subset of threats: we inquire the literature, and select (Sect. 2.1) the vulnerabilities that are expected to be a long-term threat not easily mitigated. With the selection of threats at hand, we analyze (Sect. 2.2) their packet-based forms, to construct flow-based signatures. The signatures are implemented and tested on flows collected on two production networks: (1) the National Research and Educational Network (NREN) CESNET, with 8 vantage points, totalling 2.5G of flows (87G packets, 81.2Ti bytes); and (2) the campus network UTNET, with a single vantage point, with 2.2G of flows (158.6G packets, 140.7Ti bytes).

2.1 Threat Selection Process

The comprehensive overview in [3] functions as a starting point in our selection process. In that paper, Tables II and III list Security Vulnerabilities and Privacy Vulnerabilities, respectively, indicating the origin of each threat. **Step 1:** We only consider threats originating from the *design* of IPv6, and not any threats based on *implementation* or *configuration* mistakes. We continue by looking at Table V of that same paper, which is a matrix linking threats to *detective*, *preventative* and/or *reactive* countermeasures. **Step 2:** We only consider threats that have either no forms of countermeasure, or only a *reactive* countermeasure, as our goal is detection of attacks. Lastly, we rule out threats that are not actually in IPv6 itself, but merely in other (supporting) protocols. **Step 3:** Dismiss threats based on DNS and ICMP6.

2.2 Threat Analysis Process

For each threat, the following steps are carried out:

1. At the packet-level, pinpoint the protocol fields and their respective values that make up the essence of the vulnerability.
2. Determine if the essential features found in the previous step are still available in the aggregated form (flow level). **N.B.:** availability of these features depends on which Information Elements are exported by the flow exporter. Furthermore, the flow cache should in some cases use these fields in its cache key, in order to distinguish and export separate flow records. More details on this follow in Sect. 3.
3. If an attack is not distinguishable based on information of a single flow, determine the relationship between malicious flows, as well as the relationship between the malicious and benign flows.
4. Formalize a signature based on the previous two steps, resulting in a per-attack detection approach.

3 Attack Signatures

Our selection process described in Sect. 2.1 yields six attacks (Table 2), categorized as *covert channels* (exfiltration of information), *DoS* attacks (aiming to overload and impair functioning of systems) and *middlebox evasion* (getting around *e.g.* firewalls). The constructed signatures, along with their formal explanation, are listed in Table 1. Note that we describe signatures **from the perspective of the collector**, not aggregation by the flow cache on the exporter.

The Denial of Service (DoS) signatures have two variants: the *multi-flow* kind describes an attack where a large number of destination addresses is generated randomly, as opposed to the kind where a single destination address is used. Naturally, different destination addresses lead to different flow records, and therefore different signatures.

Table 1. Signatures and notation explanation

f_i	Field in packet, *e.g.* Source Address	$5t$	Shorthand for the 5-tuple flow-key
$\{f_1, \ldots f_n\}$	Flow-key based on fields $f_1 \ldots f_n$	FL	Flow Label (IPv6 header field)
$\#$	Number of flows for flow-key or set	TC	Traffic Class (IPv6 header field)
ppf	Packets per flow	pr_n	Protocol Number n
$pps(S)$	Packets per second in flow set S	τ	Threshold, relative to context
(FK|F+)	Set of flows aggregated on FK filtered on one or more filters F		
F	Selection filter, *e.g.* $ppf = 1$ for flows with a single packet, or pr_0 for Protocol 0		

Flow Label Covert Channel	$\#(\{FL, 5t\} \mid FL > 0, ppf = 1) - \#\{5t\} > \tau_{flow\_diff}$
Traffic Class Covert Channel	$\#(\{TC, 5t\} \mid TC > 0, ppf = 1) - \#\{5t\} > \tau_{flow\_diff}$
Multi-flow Flow Label DoS	$S = (\{src\_ip\} \mid FL > 0, ppf = 1)\ ,\ pps(S) > \tau_{pps}$
Multi-flow Fragmentation ID DoS	$S = (\{src\_ip\} \mid pr_{44}, ppf = 1)\ ,\ pps(S) > \tau_{pps}$
Multi-flow Hop-by-Hop DoS	$S = (\{src\_ip\} \mid pr_0, ppf = 1)\ ,\ pps(S) > \tau_{pps}$
Flow Label DoS	$\#\{FL, 5t\} - \#\{5t\} > \tau_{flow\_diff}$
Fragmentation ID DoS	$S = (\{5t\} \mid pr_{44}, ppf > \tau_{ppf})\ ,\ pps(S) > \tau_{pps}$
Hop-by-Hop DoS	$S = (\{5t\} \mid pr_0, ppf > \tau_{ppf})\ ,\ pps(S) > \tau_{pps}$
Fragmentation Overlap	$\{5t \mid 0 < \text{fragMinOffset} \leq 20\}$

An overview of requirements for flow exporters is presented in Table 2. These requirements include certain fields to be incorporated in the flow cache key (distinguishing flows on those fields), and a new IPFIX Information Element to be implemented. Note that not all IANA assigned fields are exported per se.

Table 2. Flow record requirements for implementation of signatures

Threat	Flow key	IANA	New IE
Flow Label CC	$\{\mathbf{FL}, 5t\}$	`id31`	
Traffic Class CC	$\{\mathbf{TC}, 5t\}$	`id5`	
Flow Label DoS	$\{\mathbf{FL}, 5t\}$	`id31`	
Fragmentation ID DoS	$\{5t\}$	`id4`, `id54`	
Hop-by-Hop Option DoS	$\{5t\}$	`id4`	
Fragmentation Overlap	$\{5t\}$	`id4`	**minFragOffset**

4 Evaluation of the Signatures

The proposed signatures are evaluated on real production traffic, in which we inject generated attacks. As the DoS attacks could harm the routers, a safe number of packets is used, likely lower than a real attack but still allowing verification of our signatures. We describe the generated attacks, and discuss the performance of the signatures with respect to both these attacks and the production traffic, below.

Generated Attacks:

Flow Label and Traffic Class Covert Channels: Sending 100, 500, 1000 packets, within a 5 min time-frame, towards a single host.

Flow Label, Fragmentation ID, Hop-by-Hop Option DoS: Sending 500 packets at line rate, towards a single host; Sending 500 packets at line rate, towards randomly generated hosts in a /64 network.

Fragmentation Overlap: Sending flows of 2, 10, 20 packets, with second packet offsets of 1, 4, 10, 20 towards a single host.

Performance:

Flow Label Covert Channel: The flow records related to the covert channel are successfully distinguished, using a threshold of $\tau_{pkt} = 50$. No other positives were found in the dataset, meaning the signature has a low false positive rate but possibly a non-zero false negative rate.

Traffic Class Covert Channel: The Traffic Class can hold different values within a single flow, and we do observe this in production traffic. Most commonly, these are a zero and a non-zero value: including the TC-field in the aggregation thus results in two flows. Using $\tau_{fl} = 10$, *i.e.* marking flows with 10 or more different Traffic Class values as attacks, the signature distinguishes all the injected attacks from the production traffic. Similar to the Flow label Covert channel, no other positives where marked, pointing out a low false positive rate but a possible non-zero false negative rate.

Flow Label Flood: Detection of a Flow Label flood to a single destination address is similar to detecting a Flow Label covert channel, thus results are equivalent. Distinguishing the covert channel from the DoS attack is challenging. Multi-flow signature has a false positive rate, albeit because it marks other threats and not benign traffic. For example, a SYN scan has, on the flow level, vast similarities when compared to the flow label flood attack: a large number of end hosts is being connected to from a single source address, with every initiated connection having a new (thus different) Flow Label.

Hop-by-Hop Flood: As the Hop-by-Hop Options are not widely used (most of it is link local traffic, with only one or two packets per flow), simplistic thresholds for detection work: $\tau_{ppf} = 10$ suffices. This means scalable detection without the need for extra Information Elements or extra processing at the exporter is trivial. A possible form of false positives exists however, as we observed two times on the

NREN: ping sweeps with Hop-by-Hop Options match this signature. Marking the spread version of the attack is successful, without any other positives.

Fragmentation Flood: Detection of flooding based on the Fragmentation ID has several caveats. By definition, a flow with fragmented packets consists of more than one packet, but an exception of this characteristic are the atomic fragments. Signatures based on fragmented but single packet flows therefore yield false positives. As the sending rate and number of sent packets are crucial in the success of a flooding attack, we can choose thresholds that eliminate these false positives: $\tau_{pps} = 5000/s$, $\tau_{ppf} = 200$. Our attacks are identified without any other flows being marked, again pointing out a low false positive rate but a possible false negative rate. The case where destination addresses were generated and the flooding attack was hidden in a large number of different 5-tuple flows, is successfully detected.

Fragmentation Overlap: The approach based on *fragMinOffset* marks all our injected attacks. The lowest value observed in the production traffic was 64, so no positives other than our injected attacks were marked.

5 Conclusions

IPv6 comes with a plethora of threats specific to this new version of the IP protocol. By systematically characterising threats described in literature, we found six of these threats to be fundamental, *i.e.* based on the protocol specification and without detection approaches for attacks as of yet. In this study, we proposed flow-based signatures to perform such detection. By implementing a prototype, we proved the validity and limitations of these signatures, and defined the requirements for flow measurement equipment to allow for applying detection of attacks based on these signatures. These requirements show adaptations to flow equipment are necessary to enable for detection of these new attacks.

By deploying our prototype on two production networks and injecting attacks into the production traffic, we showed our signatures are able to successfully distinguish the attacks from benign traffic without any false negatives. We provide both the detection prototype as well as the code used for generation of attacks as free and open source software [1].

Acknowledgments. This work has been supported by the project Reg. No. CZ.02.1.01/0.0/0.0/16_013/0001797 co-funded by the Ministry of Education, Youth and Sports of the Czech Republic and European Regional Development Fund, and the Ministry of the Interior of the Czech Republic under project no. VI20162019029.

References

1. IPv6 L3 Threat Detection. https://github.com/ut-dacs/IPv6-L3-threat-detection/
2. Hofstede, R., Hendriks, L., Sperotto, A., Pras, A.: SSH compromise detection using NetFlow/IPFIX. ACM SIGCOMM CCR **44**(5), 20–26 (2014)
3. Ullrich, J., Krombholz, K., Hobel, H., Dabrowski, A., Weippl, E.R.: IPv6 security: attacks and countermeasures in a nutshell. In: USENIX WOOT (2014)