An Observation-Based Middlebox Policy Taxonomy

Korian Edeline, Benoit Donnet
University of Liège – Montefiore Institute
Belgium
firstname.lastname@ulg.ac.be

ABSTRACT

Recent years have seen the rise of middleboxes, such as NATs, firewalls, or TCP accelerators. Those middleboxes play an important role in today's Internet, including enterprise networks and cellular networks. However, despite their undisputable success in modern network architecture, their actual impact on packets, traffic, and network performance is not that much understood. In this paper, we propose a path impairment oriented middlebox classification that aims at categorizing the initial purpose of a middlebox policy as well as its potential complications.

CCS CONCEPTS

• **Networks → Middle boxes / network appliances**; *Layering*; *Network measurement*;

KEYWORDS

middleboxes, tracebox, taxonomy, path impairment

ACM Reference Format:
Korian Edeline, Benoit Donnet. 2017. An Observation-Based Middlebox Policy Taxonomy. In *Proceedings of ACM CoNEXT Student Workshop, Seoul, Korea, 2017 (CoNEXT'17),* 3 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Nowadays, the standard and well-known description of the TCP/IP architecture (i.e., the end-to-end principle) is not anymore applicable in a wide range of network situations. Enterprise networks, WiFi hotspots, and cellular networks usually see the presence of *middleboxes* [1] as being part of the network architecture in addition to traditional network hardware [16].

There is a wide range of middleboxes, going from "simple" NAT to complex system that can potentially modify headers beyond IP. Unfortunately, it has been shown that middleboxes have a negative impact on the TCP protocol (and its extensions) evolution [10, 11]. Large-scale studies of middleboxes deployment is therefore vital for the transport protocols field. However, the majority of existing studies of middlebox deployment rely on having access to configurations from enterprise networks or ISPs, which greatly reduce their scope [3, 16, 20]. Moreover, there is no rigorous behavioral middlebox classification according to their effects on packets, on traffic, or on the network quality experienced by users.

In this paper, we advocate for an *observation-based middlebox policy taxonomy*, that aims at categorizing the initial purpose of a middlebox policy as well as its potential unexpected complications. To achieve this, we run one-sided active network measurement tools on as many paths as possible: `tracebox` [4, 7, 17, 21], an extension to `traceroute` [19] that infers and locates in-path modifications, `PATHSpider` [12, 13, 18], a tool that performs one-sided A/B testing of transport-layer features to highlight in-path impairments, and `copycat` [8], a tool that evaluates experimental protocols' deployability, by comparing their processing by the network to TCP.

Then, we aggregate collected observations in a path transparency observatory (*PTO*) [15], with the aim of building an Internet-wide view of what middleboxes do to packets, on which Internet paths, and to separate low-level data handling and high-level analysis. We use this as a substrate for classifying middlebox policies.

2 TAXONOMY

From an open dataset composed of 518 million `tracebox` probes on IPv4 wired networks [2, 7], we make an attempt at categorizing inferred middlebox behavior [1] by proposing a path-impairment oriented middlebox policy taxonomy [5, 6]. As illustrated in Fig. 1, our taxonomy is based on three meta-categories that compose a policy: (*i*) **Capability**, *what* the policy expects to achieve, its purpose; (*ii*) **Action**, *how* the policy tries to achieve its goals, the fate of a packet crossing a middlebox that implements this policy; (*iii*) **Complication**, the potential resulting path connectivity *deterioration.*

We consider two basic kinds of *Actions*: *Drop* and *Rewrite.* This aspect is decisive because policies that apply different

[1] We are in the process of including the other datasets.

CoNEXT'17, 2017, Seoul, Korea

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

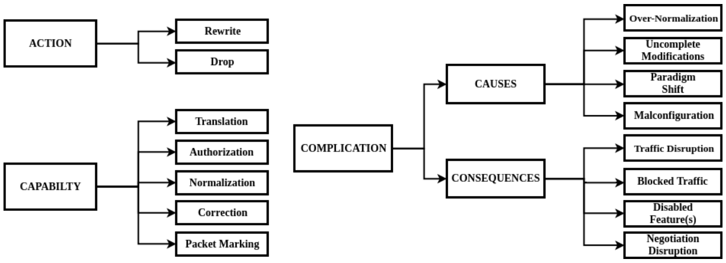

Figure 1: A middlebox policy taxonomy.

Capability		**Complication causes**	
Translation	1.3%	Malconfiguration	2.9%
Normalization	41.0%	Incomplete modifications	56.1%
Correction	54.8%	Over-Normalization	41.0%
Packet marking	2.4%	**Complication consequences**	
Authorization	NA	Traffic disruption	57.2%
Unknown	0.5%	Blocked traffic	1.3%
		Disabled features	41%
		Unknown	0.5%

Table 1: Capabilities and Complications distributions.

actions will more likely cause different types of network dysfunctions.

Then, we consider five categories of *Capabilities* (i.e., a middlebox basic feature that can be configured to enforce a policy).

Translation capabilities, that perform dynamical mapping of certain fields of a flow packets between two networks in order to be understood by each one of them (e.g., NATs). *Authorization* capabilities, that are implemented by middleboxes that discard a flow if it meets certain criterions (e.g., filtering, TCP window-checking). *Normalization* capabilities, that transform a flow by modifying fields or options in the transport header, to comply to a network policy. For example, middleboxes that limit a protocol features to a restricted subset to prevent the use of unwanted features. *Correction* capabilities, that aims at fixing endpoint implementations by transforming flows. For example, sequence number randomness. And *Packet Marking* capabilities (e.g., `Differentiated Services Code Point (DSCP)`, `Explicit Congestion Notification (ECN)`), that are normally not considered middlebox behavior but our observations showed cases of erroneous marking, via the legacy `IP ToS` field, encroaching on the `ECN` bytes, and defective `ECN` implementations.

We categorize *Complications* caused by middlebox policies by examining (*i*) their technical *Causes*, manufacturers and policy designers fundamental errors or deliberated choices and (*ii*) their *Consequences*.

Over-normalization refers to a policy that limits protocol features and options to a restricted subset. This type of behavior may constrains the design of new extensions [11], or limit protocol performance by preventing the usage of its entire capabilities, or by taking drop decisions.

Incomplete modifications refers to policies that fail to ensure completeness of their modification(s). When middleboxes modify a specific protocol field but not other semantically related fields, allowing modified data alongside unmodified data. They may fail to identify all related fields or simply neglect them for performance concerns.

A *Paradigm shift* happens when both ends running a protocol still assume 2-way peering relationships. Middleboxes cause both endpoints to undergo a paradigm shift *de facto* to `n`-way peering relationships [14]. As many mechanisms are not designed to handle this new paradigm, errors may occur. When both ends try to share state related data or to negotiate capabilities, this phenomenon may, in certain scenarios, put both ends in conflicting states [10].

Finally, we identify as *Malconfigurations* vendor implementation or design errors in capabilities, leading to faulty middlebox policies.

Among the *Consequences* (i.e., what both ends actually experience), *Traffic disruption* policies impairs performances, they may, for example, interfere with control data rendering it useless, or reduce bandwidth. *Blocked traffic* policies, that can be either explicit (sending TCP `RST` packet) or implicit (dropping packets). Middlebox policies may also prevent the use of features considered unknown or unsafe. If done symmetrically, consequences are limited to the inability to use them; it is a *Feature-disabling* policy. If the modifications are asymmetric and the negotiation is not resilient enough, a *Negotiation disruption* policy may fail to disable the feature correctly and lead to inconsistent protocol states [10].

Capabilities and consequences distributions are displayed in Table 1. We did not observe authorization policies because they are mostly out of scope for the measurement techniques that we are currently using. Our probes inferred few translation capabilities because NATs are required to be invisible to `tracebox` [9]. However, we recently developed a technique to detect them regardless [21]. We also observed that few packet marking capabilities are prone to create transport-level impairment. Finally, we observed large proportions of normalization and correction middleboxes, with an advantage for the second category that can be explain by the fact that middleboxes implementing such policies tends to be located closer to access networks, which affects more paths in our dataset. A second explanation is that normalization policies have more incentives to be invisible because they reveal pieces of information about AS traffic engineering.

Finally, we find that the most common impairments are traffic disruption and disabled features, and that they are caused by, respectively, incomplete packet modifications and over-normalization.

3 NEXT STEPS

In the future, we plan to augment the *PTO* with observations from our other network measurement tools (`PATHSpider` and `copycat` [8, 13]) and our NAT-detection methodology [21], and to use them to refine the taxonomy. We are also planning to implement middleboxes classes in a simulator for offline protocol testing.

Furthermore, we plan to add operational middleboxes characteristics. In particular, we are interested in middlebox prevalence (e.g., if a firewall is set up, does all traffic go through that firewall?), persistence over time (e.g., is a middlebox up and running all the time, or do we observe any dynamics as for IP networks?), and middlebox position in its AS topology (e.g., at the border or in the core) [7].

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688421. The opinions expressed and arguments employed reflect only the authors' views. The European Commission is not responsible for any use that may be made of that information.

REFERENCES

[1] B. Carpenter and S. Brim. 2002. *Middleboxes: Taxonomy and Issues.* RFC 3234. Internet Engineering Task Force.

[2] MAMI Consortium. 2016. Path Transparency Observatory. (2016). https://observatory.mami-project.eu/

[3] L. D'Acunto, N. Chiluka, T. Vinò, and H. J. Sips. 2013. BitTorrent-Like P2P Approaches for VoD: a Comparative Study. *Computer Networks (COMNET)* 57, 5 (April 2013), 1253–1276.

[4] G. Detal, b. Hesmans, O. Bonaventure, Y. Vanaubel, and B. Donnet. 2013. Revealing Middlebox Interference with Tracebox. In *Proc. ACM Internet Measurement Conference (IMC).*

[5] K. Edeline and B. Donnet. 2015. On a Middlebox Classification. In *IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI).*

[6] K. Edeline and B. Donnet. 2015. Towards a Middlebox Policy Taxonomy: Path Impairments. In *Proc. 7th IEEE International Workshop on Science for Communication Networks (NetSciCom).*

[7] K. Edeline and B. Donnet. 2017. A First Look at the Prevalence and Persistence of Middleboxes in the Wild. In *Proc. International Teletraffic Congress (ITC).*

[8] K. Edeline, M. Kühlewind, B. Trammell, and B. Donnet. 2017. copycat: Testing Differential Treatment of New Transport Protocols in the Wild. In *Proc. ACM/IRTF/ISOC Applied Networking Research Workshop (ANRW).*

[9] S. Guha, B. Ford, S. Senthil, and S. Pyda. 2009. *NAT Behavioral Requirements for ICMP.* RFC 5508. Internet Engineering Task Force.

[10] B. Hesmans, F. Duchene, C. Paasch, G. Detal, and O. Bonaventure. 2013. Are TCP Extensions Middlebox-Proof?. In *Proc. Workshop on Hot Topics in Middleboxes and Network Function Virtualization (HotMiddlebox).*

[11] M. Honda, Y. Nishida, C. Raiciu, A. Greenhalgh, M. Handley, and H. Tokuda. 2011. Is It Still Possible to Extend TCP. In *Proc. ACM Internet Measurement Conference (IMC).*

[12] I. R Learmonth, A. Lutu, G. Fairhurst, D. Ros, and Ö. Alay. 2017. Path transparency measurements from the mobile edge with PATHspider. In *Proc. Network Traffic Measurement and Analysis Conference (TMA).*

[13] I. R Learmonth, B. Trammell, M. Kühlewind, and G. Fairhurst. 2016. PATHspider: A tool for active measurement of path transparency. In *Proc. ACM/IRTF/ISOC Applied Networking Research Workshop (ANRW).*

[14] M. A. Lemley and L. Lessig. 2000. *The End of End-to-End: Preserving the Architecture of the Internet in the Broadband Era.* Technical Report 2000-19. University of California at Los Angeles.

[15] S. Neuhaus, R. Münter, K. Edeline, B. Donnet, and E. Gubser. 2016. Towards an observatory for network transparency research. In *Proc. ACM/IRTF/ISOC Applied Networking Research Workshop (ANRW).*

[16] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. 2012. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service. In *Proc. ACM SIGCOMM.*

[17] V. Thirion, K. Edeline, and B. Donnet. 2015. Tracking Middleboxes in the Mobile World with TraceboxAndroid. In *Proc. 7th International Workshop on Traffic Monitoring and Analysis (TMA).*

[18] B. Trammell, M. Kühlewind, P. De Vaere, I. R Learmonth, and G. Fairhurst. 2017. Tracking transport-layer evolution with PATHspider. In *Proc. ACM/IRTF/ISOC Applied Networking Research Workshop (ANRW).*

[19] V. Jacobson et al. 1989. *traceroute.* man page. UNIX.

[20] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. 2011. An Untold Story of Middleboxes in Cellular Networks. In *Proc. ACM SIGCOMM.*

[21] R. Zullo, A. Pescapé, K. Edeline, and B. Donnet. 2017. Hic Sunt NATs: Uncovering Address Translation with a Smart Traceroute. In *Proc. IEEE/IFIP Workshop on Mobile Network Measurement (MNM).*