NetAssay: Providing New Monitoring Primitives for Network Operators

Sean Donovan
Georgia Institute of Technology
sdonovan@cc.gatech.edu

Nick Feamster
Georgia Institute of Technology
feamster@cc.gatech.edu

ABSTRACT

Home and business network operators have limited network statistics available over which management decisions can be made. Similarly, there are few triggered behaviors, such as usage or bandwidths cap for individual users, that are available. By looking at sources of traffic, based on Domain Name System (DNS) cues for content of particular web addresses or source Autonomous System (AS) of the traffic, network operators could create new and interesting rules for their network. NetAssay is a Software-Defined Networking (SDN)-based, network-wide monitoring and reaction framework. By integrating information from Border Gateway Protocol (BGP) and the Domain Name System, NetAssay is able to integrate formerly disparate sources of control information, and use it to provide better monitoring, more useful triggered events, and security benefits for network operators.

Keywords

Network monitoring, network management, software-defined networking

1. INTRODUCTION

Network management is a complex area. On stub networks, where the network is only a client to other networks, monitoring user behavior is limited. Bandwidth monitoring at the port or address level is among the more complex operations that are available. Similarly, triggered behaviors such as usage caps are limited. More complex behavior is available, primarily in highly specialized security appliances, however these are not general purpose devices.

What is needed is a generalized system that can monitor network activity and produce behaviors based on high level policies. For instance, a network operator would like to know how much video traffic each user is consuming. Right now, there is no easy way of doing this.

NetAssay is being developed to make it easy for network operators to implement such policies. In the video bandwidth example, NetAssay needs to decide what network flows are video traffic. Traffic coming from `googlevideo.com` will be classified as video, as could any traffic associated with AS 2906, Netflix's AS. Identifying the user, at least on a stub network, can be done a number of ways, such as statically assigned IPs or MAC addresses. A policy for this could be expressed as:

```
video-count-Bob = count_bytes(3600,['src-ip'])
video-count-Bob.register_callback(check_video_count)
video-from-Bob = match(srcip = 198.51.100.8) &
                 (matchAS(2906) | matchClass('video'))
pol = video-from-Bob >> video-count-Bob
```

More interesting triggered behaviors are also possible. Continuing the above example, being able to react to Bob's excessive video consumption is desirable. For instance, being able to redirect Bob's video traffic through a rate limited link could be a useful policy to implement.

As a further, more security oriented example, a network operator has blacklists of ASes and domains that are associated with malicious behavior, but do host some legitimate content that should still be accessible. In this case, if a connection was destined for an IP associated with a potentially malicious AS, the network could automatically redirect traffic to a virus scanning appliance, rather than needing to send all traffic through the appliance. Another use would be to set a limit for amount of data that would be sent to a blacklisted AS before blocking connections, due to suspected bad behavior.

Similar triggered behaviors have been described in prior works [5][1], however this has been limited to network security security devices.

2. DESIGN

NetAssay is SDN-based system running on Pyretic [3], an easy to use and extend SDN programming language. NetAssay is divided into two sections, as can be seen in Figure 1. These are the monitoring modules, two of which are described below, that provide traffic identification and monitoring information to the main control module (MCM). The MCM is responsible for combining the results from the monitoring modules, and triggering the reactive policies defined by the network operator.

2.1 Monitoring Modules

The monitoring modules perform two major functions: identifying flows that belong to a particular category and monitoring information related to that category. To identify flows, the modules can use information gleamed from

SIGCOMM'14, August 17–22, 2014, Chicago, IL, USA.
ACM 978-1-4503-2836-4/14/08.
http://dx.doi.org/10.1145/2619239.2631451.

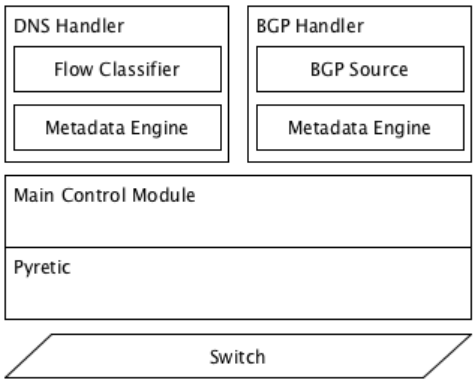

Figure 1: NetAssay architecture

monitoring network flows or from external data sources. The monitoring modules are designed such that new modules can be added in the future.

2.2 DNS and BGP Handlers

For the initial implementation, there are two modules being developed. Both are similar in design, but use different sources of information. The first, a DNS-based handler, will be described in detail, while the BGP-based handler will be quickly described.

The DNS handler which contains two parts, a flow classifier based on part of FlowQoS [4] and a metadata engine. To perform flow classification, two pieces of information are needed: URL-to-classification mapping and IP-to-URL mapping. The URL-to-classification mapping is mentioned above where `googlevideo.com` maps to 'video'. This is a manual lookup, implemented as regular expression matching. The IP-to-URL mapping is based on monitoring DNS responses. At initialization, rules are installed at all ingress network devices to make a copy of DNS responses, forward one to the actual destination, and forward the second to the DNS handler. The flow classifier then extracts the domains and associated IP addresses, and uses the URL-to-classification mapping to determine what type of traffic is associated with a particular IP address. The metadata engine handles policy implementation for DNS-based classification. The metadata engine is responsible for breaking down the high-level policies passed form the MCM into the constituent components, and handling any changes that my occur dynamically (*i.e.*, when a new DNS response comes in, or DNS response time-to-live (TTL) is reached). It is also responsible for aggregating data, such as aggregating the number of bytes transferred by video sources.

To go over the example above, if during the example above a DNS response from `googlevideo.com` with associated IP of 198.51.100.8 was seen, the URL will be classified as video, and, as such, traffic coming from 198.51.100.8 would be classified as video. This will cause the `matchClass` policy to generate a new `match` action to run in parallel with any preexisting actions associated with video traffic.

To allow for rules based on traffic related to a particular AS, BGP information is necessary. NetAssay uses an external source of BGP information. This could be a connection to a local BGP server, or could be a static database of known

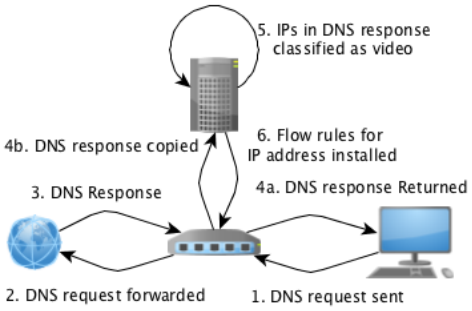

Figure 2: Behavior of NetAssay's DNS handler classifying a video flow.

routes. The BGP information source plays a similar role to the Flow Classifier in the DNS handler: it provides a lookup source for AS routes, and, more importantly, ranges of addresses controlled by a particular AS.

3. PROGRESS AND FUTURE WORK

Work on NetAssay is ongoing. Design work on the two handlers is complete, with implementation and testing ongoing. Triggered behaviors is the current area of focus. Further simplification, such as simplifying the policy in the introduction to be a single line, is desired.

Future work, beyond finishing implementation and testing, primarily focuses on optimization. Proactive DNS look ups from various public DNS sources provides a broader data set. Additionally, handling conflicts between handlers' policies needs to be addressed, both with data flows and potential 'double counting' between multiple handlers.

Further work incorporating this system into an Internet Exchange Point, particularly in the form of SDX [2], is a natural next step

4. REFERENCES

[1] J. R. Ballard, I. Rae, and A. Akella. Extensible and scalable network monitoring using opensafe. *Proceedings of USENIX Internet Network Management Workshop/Workshop on Research on Enterprise Networking*, 2010.

[2] A. Gupta, L. Vanbever, M. Shahbaz, S. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. Sdx: A software defined internet exchange. *Proceedings of the ACM SIGCOMM 2014 conference*, 2014. To Appear.

[3] J. Reich, C. Monsanto, N. Foster, J. Rexford, and D. Walker. Modular sdn programming with pyretic. *USENIX ;login,*, 38(5):128–134.

[4] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-Q. Song. Flowqos: Qos for the rest of us. *HotSDN 2014*, 2014. To Appear.

[5] S. Shin and G. Gu. Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks. *20th IEEE International Conference on Network Protocols*, 2012.