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

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 in a month, and, if they reach a specified limit, limit their bandwidth for the rest of the month. 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 `googlevideocontent.com` will be classified as video, as could any traffic associated with AS 2906, Netflix's AS.

More interesting triggered behaviors are also possible. For instance, 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.

1.1 Behavioral Queries

Before going into the design, a few useful examples follow. This is in Pyretic syntax [3], where `>>` is sequential composition (operations run in order) and `+` is parallel composition (operations run in parallel).

```
traffic_in_bytes(to = 'BobsPC',
                 content = 'video',
                 timespan = 'monthly') > 10**9 >>
    rate_limit(to = 'BobsPC',
               content = 'video',
               bandwidth = 65536)
```

The example above is a variation on the video bandwidth limitation example. In this case, we only care if Bob's PC is downloading more then 10 gigabytes of video traffic in a month. If it does reach this limit, then we want to limit the bandwidth to 64 kilobytes per second.

```
match_AS(pkt.srcip, AS = '12345')) >>
    drop()
```

The second example is simpler. We are see if a packet came from a AS 12345. If so, we want to drop it. Implicitly, if we do not drop a packet, it will be forwarded.

2. DESIGN

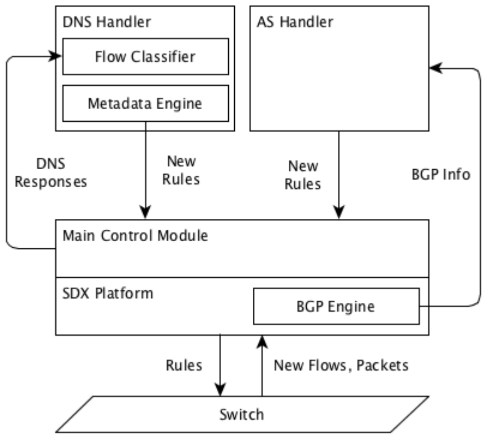

Figure 1: NetAssay architecture

NetAssay is SDN-based and workson top of the SDX platform [2]. This provides easy to understand syntax for network operators. NetAssay is divided into three parts, as can be seen in Figure 1. There are two monitoring modules, 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 implementing the reactive policies defined by the network operator.

2.1 DNS Handler

The DNS handler contains two parts, a metadata engine and a flow classifier based on part of FlowQoS [4]. At startup, the DNS handler creates a rule in 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. This is send to the flow classifier. The flow classifier parses the DNS packet, extract the doamins and associated IPs, and classifies traffic from the IPs based on the domain name. For instance, if a DNS response from `googlevideocontent.com` with associated IP of 198.51.100.8 was seen, traffic coming from 198.51.100.8 would be assumed to be video until the DNS reponse time-to-live (TTL) was reached.

The metadata engine handles policy implementation for DNS-based classification. Continuing with our video example, if a policy is written to calculate how much video traffic is coming into a network, the metadata engine is responsible for breaking down the high-level policy passed from the MCM into the constituent components. That is, for each new IP address associated with the video classification, it is responsible for creating byte-counter rules for each IP, then agregating the values reported, and returning this information to the MCM.

2.2 AS Handler

As NetAssay is based on the SDX platform which contains a BGP-engine, we have access to BGP routes. In the case where a network operator wants to send all traffic from a particular AS to a virus scanner appliance, the AS handler will identify which IP addresses are associated with the specified AS, and return these to the MCM. The MCM can then route the traffic vide the virus scanner appliance.

Many networks, in particular home or small business networks, BGP information is not accessible. As such, this handler can be seen as optional.

2.3 Data Flow

Policy is established at startup of NetAssay, so anything that happens subsequently is dynamic. In particular, the DNS and BGP bindings will change, expire, and be repopulated regularly. We will now describe part of the behavior of the first example, limiting Bob's video consumption. We are only going to discuss the actions of the DNS handler, but the actions of the AS handler will be substantially the same.

When a new DNS response comes in, the response is parsed and classified. If it is classified as video, the DNS metadata engine will create new rules that establishes byte counters for Bob's PC to all traffic coming from the IP addresses associated with the video site. It does not matter if Bob was associated with the DNS request, the rules would be installed regardless.

There are likely multiple addresses associated with video traffic, and likely as many counters. These counters will need to be accumulated, which is also handled by the DNS metadata engine. The total of the counters is sent to the MCM, which, if the total has reached 10 gigabytes for the month, will create a `ratelimit` rule and write the rule to the closest switch to Bob's PC.

3. PROGRESS AND FUTURE WORK

Work on NetAssay is ongoing. Design work is progressing significantly, however there are some language issues to be worked out. The DNS flow classifier is complete, however the classification data set is small and video-focused at the moment.

There are some inherent limitations, but will be resolved thanks to progress on other project. In particular, SDX is limited to a single switch right now, however this may be a minor issue due to ongoing SDX development.

Of note is that the DNS flow classifier is the only module that does packet inspection. Other packets are looked at

Future work, beyond finishing currently planned work, is to expand upon the proactively installed rules. Since flows can be classified to pass through a security appliance if they're coming from suspicious sources, scalability studies are possible. Proactive DNS lookups Handling conflicts

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.