FireCircle: GRNET’s approach to advanced network security services’ management via bgp flow-spec and NETCONF

Leonidas Poulos
Network Applications Developer
(leopoul@noc.grnet.gr)

Michalis Mamalis
Network Engineer
(mmamalis@noc.grnet.gr)

Andreas Polyrakis
NOC Technical coordinator
(apolyr@noc.grnet.gr)

Greek Research & Technology Network (GRNET) NOC
56, Mesogeion Ave. 11527 Athens, Greece

Abstract

Security, and in particular reactive protection against network attacks and anomalies, is a major concern of modern networks. This paper presents “FireCircle”, a novel security provisioning service that allows the administrators of a downstream network to inject firewall-like rules at their upstream level and mitigate attacks there; under certain circumstances these rules can be further propagated so that attacks are mitigated even closer to their sources. FireCircle has been designed and implemented by GRNET Network Operations Center, using hardware that supports RFC5575 (a.k.a BGP flow-spec), open-source tools and platforms and software developed in-house. The BGP nature of the service makes it quick, scalable, manageable, secure and multidomain. FireCircle adds to this mechanism a web administrative and user interface, a programmatic API, strong and scalable authentication mechanisms (Shibboleth), advanced and flexible authorization schemes and self-protection anti-measures, all packed under an open-source licence. These features allow the service not only meet its primary goals, that is the easy mitigation of attacks by personnel with medium networking expertise and low-end equipment, but also open potentials for integration with network anomaly detection tools and promote synergies between peering domains or cooperating security teams.

This paper covers the aforementioned points, providing details about the architecture, the implementation, the deployment on GRNET production network and the future plans of the service. Also, this paper discusses how FireCircle can be extended into a multidomain environment, taking the extended GEANT community as a case study.

Keywords: bgp flow-spec, security, DDOS, multi-domain, NETCONF

Paper Type: Research

1. Introduction

Throughout the world, governments, defence industries, and companies in finance, power, and telecommunications are increasingly targeted by overlapping surges of cyber attacks. The number of attacks is now so large and sophisticated that many organizations are having trouble mitigating them. To address many of these issues imposed by modern network threats in a reactive manner, we present a new security provisioning service based on bgp flow-spec as described in RFC 5575 [1]. This service, named “FireCircle” is designed and implemented in-house by GRNET Network Operations Center using free open-source tools. GRNET (Greek Research and Technology Network) provides Internet connectivity and services to all Greek Universities and academic and research institutes.

2. Background and Motivation

Attackers typically use compromised systems, “zombies”, that are relatively easy to control. In large numbers, zombies, pose a serious threat to modern networks. Take for example an army of zombies that numbers 500 bots originating from different domains as depicted in Figure 1. Such botnets can easily produce 1Gbps of attack traffic, even with bots behind 2Mbps DSL lines. The volumes of such DDOS attacks often reach magnitudes that significantly affect not only the host under attack but the overall performance of the hosting network as well, as they consume capacity and overwhelm firewalls and routers. On the other hand, other modern attack techniques like reflective Amplified DNS spoofing [2][3] do not leave a trace of traffic spikes for the network administrators to follow; yet they can be
equally catastrophic.

**Figure 1. An army of zombies attacking a domain**

The traditional approach is to deploy old style access lists either on or close to the host under attack. Unfortunately, this approach does not prevent the waste of valuable resources, such as uplink bandwidth; and it can still overwhelm low-end equipment (router, firewall, etc) before the mitigation point. As an alternative, the access-lists can be placed further upstream; however this usually demands human intervention and, thus, big installation times and high administrative cost.

A first attempt to automate attack mitigation was made with Remote Triggered Black-Hole (RTBH) [4]. Using a well known protocol, BGP, as a security tool, the receiving router translates a BGP community into a discard Next-Hop [5]. This approach solves the issue of automatic propagation, but introduces new drawbacks itself: The filters can only match traffic on per-IP level, and the only available action was to drop traffic, which in certain cases would only complete the attack by making the victim unreachable to the entire Internet. Moreover, the RTBH rules are installed directly on the global routing table, which creates routing and security issues and makes management and troubleshooting more difficult.

An enhancement to existing approaches is the deployment of a security mechanism that would allow firewalling rules to be propagated to different domains in a manner independent of routing, while allowing for a set of matching and filtering actions that could match a large range of possible attack patterns and required actions. The protocol that addresses those issues is BGP flow specification (bgp flow-spec) as described in [1].

Flow specification provides a method through which traffic flow specifications can be distributed between BGP peers, using a new BGP NLRI encoding format. In this way, each BGP speaker is allowed to propagate information that describes a traffic flow and the desired actions for this flow. The receiving peer interprets the information of the flow specification NLRI into firewall filters that accurately describe traffic flows and actions in an n-tuple of match and set rules.

BGP flow specification enhances traditional RTBH in several manners: (a) it allows fine-graded match criteria (b) it is much more flexible in terms of actions (c) it uses a separate NLRI and thus flow-spec rules do not mess with the routing table. At the same time, the BGP nature of the technology inherits a number of significant properties, such as (a) a tested, scalable and flexible control plane (BGP NLRIs) (b) its well-known and trusted authorisation mechanisms (eg route-maps that control advertisements) (c) its already established trust models, peerings and experience and (d) its multidomain nature.

On the other hand, bgp flowspec requires routers that support RFC5575 and personnel with high networking expertise with access to these routers. Unfortunately neither two is easy; Customers often do not have the appropriate equipment and are not willing to configure BGP flow-spec NLRI on them. Moreover, server administrators do not have access to their domain routers anyway.

This led hardware companies in developing and providing commercial product solutions to facilitate mitigating network attacks [6]. Effective as they might be, these products are closed-source solutions
and thus cannot be extended in-house to match the needs of a demanding environment like GRNET. Motivated by this gap, GRNET decided to implement FireCircle, an open-source, open-API, user-friendly, secure, shibbolized service that will allow its customers to mitigate transient anomalies by injecting manually (via a web interface) or automatically (via an API) dynamic firewall filters directly within GRNET network.

3. FireCircle Overview

FireCircle is a security provisioning web tool that allows GRNET customers mitigate DDOS attacks and anomalies quickly and without any human intervention from GRNET NOC personnel. In addition, the tool can consume data from trusted security tools (eg IDS, honeypots) and automatically mitigate (drop or rate limit) reported anomalies. The aforementioned anomalies are mitigated upon entrance on GRNET network; this behaviour can be further enhanced by allowing the “circle” to grow in neighbouring domains, provided that they deploy equipment that supports RFC5575. All it takes is to enable the flow-spec NLRI over the existing BGP sessions. BGP provides all necessary control-plane mechanisms, including a widely accepted trust model, established peering and tested BGP filters. Eventually, BGP will propagate the flow filters and the attacks will be mitigated closer to their source.

![Figure 2. Attack mitigation](image)

Figure 2 demonstrates this concept, using GRNET as a case study: Suppose that GRNET started to exchange flow-spec NLRI with its upstream, GEANT.[7] A DDOS attack sourced behind an NREN-X, towards GRNET, would be blocked upon entrance on the GEANT network, i.e on GEANT’s port where NREN-X is connected. Taking the example one step further, in case that the NREN-X accepted flow-spec NLRIs from GEANT, the attack would be blocked even closer to the source; and all these, with a minimal administrative effort.
4. Architecture and Implementation

FireCircle is developed on the Python Django framework [8]. FireCircle’s core architecture is shown in Figure 3.

The FireCircle service is available to its users through a web interface. Users are authenticated via Shibboleth [9], and are authorized through an appropriate Entitlement attribute. Each customer is responsible to assign this attribute to the appropriate personnel on their own IdP, which disengages GRNET from user management, and at the same time provides a federated, trusted and tested authentication and authorisation mechanism. The incorporation of Shibboleth makes the platform a perfect candidate for EduGAIN [10].

For security reasons, users are only allowed to view and change firewall filters for flows that are destined to their own networks. For this reason, an additional authorisation mechanism that correlates IP space with GRNET customers has been established. This mechanism is pre-populated with data collected from RIPE and private whois servers with the help of appropriate python scripts; then prefixes can be added or removed according to needs. Eventually, the users of each Autonomous System (AS) are allowed to impose flow filters as long as their destination address belongs to the address space allocated to their own AS.

The user interface is implemented using the Django templates framework enhanced with the Javascript jQuery library [11] for rich user experience. Rules appear per domain in a table format, and filtering per name, rule details, dates and status is possible. The platform allows for creation, modification and deletion of flow rules via a wizard-like GUI as shown in Figure 4 and Figure 5. The user can create a rule based primarily on attack source address and victim’s destination address. As the tool implements the majority of bgp flow specification functionalities, the user can perform fine grain filtering by optionally selecting source and/or destination address(es), protocol type(s) and the rule action. Currently, plain users can select among discard and rate limit as rule actions whilst the administrator of the tool can select from a wider variety of actions such as sample and redirect. The user selects the expiration date of the rule which at the moment cannot exceed 7 days from the rule creation or modification date. The users along with the domains they are members of, are notified about rule
creations, modifications or deletions via email. For security reasons, each user interaction is logged and triggers a mail dispatch to the administrator of the service. Once a rule reaches its expiration date, the user is notified about the automated rule removal from the network. If the user takes no action the rule is automatically removed from the network. Optionally the user can either choose to terminate the rule prior to its expiration date, or extend its application period by altering its expiration date. Modification of rules is carried out in the same manner that creation does.

Once a rule change (addition, modification, deletion) takes place, the user web interface is automatically updated via a mechanism which is called HTTP long-polling or Comet [12]. This mechanism is incorporated in most social network applications (Facebook Chat, Twitter updates, Google Talk) and is in charge of status updates without page reloads or frequent HTTP polls. For the sake of action logging the user interface comes with a console-like dialog box as shown in Figure 6 which is also updated via the Comet mechanism.
Propagates flow-spec data to GRNET core network through eBGP sessions. BGP introduces certain important advantages compared to “traditional” alternatives such as CLI, SNMP, etc: (a) it is vendor-independent and standard (b) it does not require elevated access on the core routers and (c) protection mechanisms can be deployed. These properties make the service independent of the hardware of the core network and require minimum trust between different administrative domains (i.e., developers/operators of FireCircle and core network administrators). The existence of two (or more) eBGP peerings increases service availability, and BGP filters protect these peerings, making impossible for FireCircle to affect certain core infrastructure and non-customer IP space. These filters provide a security net that protects the core network from possible malfunction or compromise of the service.

In order to implement the BGP peering, FireCircle utilizes a dedicated hardware router – more specifically a Juniper EX4200 layer3-switch – capable of supporting BGP and flow-spec address-family. Another alternative would be to use a software bgp daemon such as ExaBGP[13], however implementation-wise the use of a cheap layer3 switch appeared simpler and more manageable. The configuration of the dedicated hardware is carried out solely via NETCONF[14]. NETCONF was chosen for being a secure management protocol with clean XML structure and a well defined request/response schema.

On the backend of the tool, an in-house developed python-to-NETCONF proxy middleware[15] translates user requests to BGP flow rules and vice-versa. BGP flow-rules are in NETCONF device configuration XML format as shown in Figure 7.

The NETCONF middleware applies the produced configuration to the hardware box via a python SSH-NETCONF client, ncclient[16], patched and now maintained by the author of this paper. Configuration retrieval from the device is also supported to allow for syncing and reconciliation. Device configuration is mapped to Python classes that can be easily distributed and reused among a variety of applications. Once a request for a new flow rule is placed, configuration is applied via NETCONF to the aforementioned hardware box and is then propagated via BGP to GRNET’s routers. If our upstream peering supported the bgp flow NLRI, that would mean that the rule would automatically be propagated to GEANT domain.

An administrative account gives the tool administrator an overview of all active rules on all domains. The administrator can create, modify or delete rules on behalf of all domains. This account can support
domains that are not currently within GRNET’s AAI scheme and thus cannot access the tool via Shibboleth.

5. Status
Currently the tool is deployed in GRNET’s production network as a stable beta and has been used to mitigate approximately 10 network attacks. Prior to production network deployment, the tool was extensively tested in GRNET’s test-bed environment. The tests were conducted emulating network attacks with iperf [17]. As shown in Figure 8, the emulated attack is mitigated within approximately 13 seconds.

![Figure 8. Attack mitigation emulation](image)

Moving to the production network showed that the average configuration application time was approximately 12 seconds. What is worth mentioning is the fact that once a user decides to apply multiple rules simultaneously, these rules are applied within the same single NETCONF session. In practice this translates to very low configuration application times, almost close to the aforementioned one of 12-13 seconds.

6. Planned Extensions
As the development of new features is ongoing a roadmap has been set to help address major needs and requirements for the tool. Currently, a REST API has been developed and is under testing that will allow rules’ creation, modification and deletion. The goal is to integrate with applications such as network anomaly detection tools and traffic flow analyzers and engage dynamically bgp flow rules in case an attack is detected. This will boost the role of FireCircle as it can become part of a robust security solution against network threats.

Monitoring is the other important feature of the tool. Currently, firewall rule application to GRNET’s devices, aka. Juniper routers, triggers the generation of a new firewall counter within the device. This counter is inaccessible via SNMP which makes monitoring of firewall rules’ counters difficult. There are two approaches to solve this issue. The first approach is to use NETCONF to gather this information. But since data acquisition from the device counters has to be carried out every few minutes, NETCONF would probably impose greater overhead compared to SNMP. Fortunately, the devices’ vendor provides an excellent workaround to acquire this information indirectly via SNMP. Juniper devices offer a user MIB which can be filled via Juniper’s scripting language Slax [18]. Currently, such a script has been developed and is under testing in GRNET’s test-bed. This script which is considered an event script, runs every five minutes, gathers the bgp flow specification firewall
counters reading and stores this information to certain user MIB values. These values can be retrieved via SNMP and eventually turn into firewall rule graphs. IPv6 support is probably one of the most desired enhancements of the service. We have contacted Juniper, which is the hardware vendor of both the hardware used in FireCircle and GRNET core routers, and we received assurance that IPv6 support is within their roadmap. Supporting IPv6 will require minimum development intervention as the current Python library used, python-ipaddr [19], is IPv6 enabled. The tool’s source code will be soon available for downloading [20] as an open source project.

7. Possible Synergies
FireCircle opens the way for a number of synergies. First of all, for peers that support RFC5575, we can start by enabling the bgp flow-spec NLRI on our BGP peerings. In this manner our filters will be propagated further – this makes a lot of sense, especially in our upstream direction. At the same time, given that our peers have a similar service, we will accept their own filters. Note that the nature of BGP allows the exchange of such filters even with networks that are not directly connected. Hence, with minimal operational overhead, such synergies will mutually benefit all peers.

Secondly, for networks that do not have such a service deployed, FireCircle is available in open-source to download and install. Such collaborations could further enhance the software and, thus, be beneficial to the service.

Last but not least, synergies with security teams and tools are envisioned. Once the REST API for the tool is available, platforms like NSHARP (developed and used by GEANT), could feed FireCircle and automatically mitigate ongoing attacks. In general, data exported by any security tool or produced by any security team could be used for this purpose. In addition, any security team could use multihop BGP peerings to export security data [21].

Finally, we believe that this service is a perfect candidate for GN3-like projects, that promote the cooperation of different network. Its multidomain nature makes it easy to adopt in an environment like GEANT, its Shibboled nature makes it a perfect candidate for eduGAIN [10], and there are is a lot of room for cooperation with the efforts in the direction of security that currently take place within GEANT.

8. Conclusions
In this paper we presented an innovative security service, namely FireCircle, that protects a domain against network attacks. The nature of the service allows for multidomain deployment with minimum overhead, and makes it perfect candidate for communities like GEANT and NRENs. What makes this service unique is its open-source nature, the shibboled user interface, the use of modern programming techniques (long polling, queuing, caching) and open protocols such as NETCONF, on top of the powerful, multidomain and secure mechanisms of BGP flow-spec. A screen cast of the tool in action is available at [22].

Authors’ short bio
Leonidas Poulopoulos received his Diploma in Electrical and Computer Engineering from the University of Patras in 2005 and his M.Sc degree on Computer Science from the Department of Computer Engineering and Informatics (University of Patras) in 2010. Currently, he is with the development team of GRNET NOC.

Michalis Mamalis received his Diploma in Electrical and Computer Engineering from the Democritus University of Thrace in 2002. His diploma thesis focused on the design and implementation of an MMIC board acting as the optical receiver part of an STM-4 optical link. His areas of interest include MPLS, routing protocols, IPv6 and policy based networking. Currently he is with the routers administration team of GRNET NOC.

Andreas Polyrakis received his Diploma from the Department of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece in 1999 and his M.Sc. degree on Computer Science from the University of Toronto, Canada in 2001. He is currently employed by GRNET as the Technical Coordinator of GRNET NOC. His areas of interest include routing protocols, IPv6, MPLS, Metro Ethernet and Cloud Computing.
References:

[17] Iperf: http://sourceforge.net/projects/iperf/
[18] Juniper Slax scripting language:
[22] GRNET’s FireCircle in action, screencast: http://www.youtube.com/watch?v=VZy7DG8sjSk