DOTS
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2015

T. Reddy
D. Wing
P. Patil
M. Geller
Cisco
M. Boucadair
France Telecom
June 29, 2015

Co-operative DDoS Mitigation
draft-reddy-dots-transport-00

Abstract

This document discusses mechanisms that a downstream Autonomous System (AS) can use, when it detects a potential Distributed Denial-of-Service (DDoS) attack, to request an upstream AS to perform inbound filtering in its ingress routers for traffic that the downstream AS wishes to drop. The upstream AS can then undertake appropriate actions (including, blackhole, drop, rate-limit, or add to watch list) on the suspect traffic to the downstream AS thus reducing the effectiveness of the attack.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2015.

Table of Contents

1. Introduction

A distributed denial-of-service (DDoS) attack is an attempt to make machines or network resources unavailable to their intended users. In most cases, sufficient scale can be achieved by compromising enough end-hosts and using those infected hosts to perpetrate and amplify the attack. The victim in this attack can be an application server, a client, a router, a firewall, or an entire network, etc. The reader may refer, for example, to [REPORT] that reports the following:

- Very large DDoS attacks above the 100 Gbps threshold are experienced.

- DDoS attacks against customers remain the number one operational threat for service providers, with DDoS attacks against infrastructures being the top concern for 2014.

- Over 60% of service providers are seeing increased demand for DDoS detection and mitigation services from their customers (2014), with just over one-third seeing the same demand as in 2013.

Enterprises typically deploy DDoS monitoring appliances that are capable of inspecting and monitoring traffic to detect potential DDoS threats and generate alarms when some thresholds have been reached. Most of these tools are offline; further steps are required to introduce online tools that would have immediate effects on traffic associated with an ongoing attack. Thanks to the activation of dynamic cooperative means, countermeasure actions can be enforced in early stages of an attack, which can optimize any service degradation that can be perceived by end users.

This document describes a means for such enterprises to dynamically inform its access network of the IP addresses that are causing DDoS. The access network can use this information to discard flows from such IP addresses reaching the customer network.

The proposed mechanism can also be used between applications from various vendors that are deployed within the same network, some of them are responsible for monitoring and detecting attacks while others are responsible for enforcing policies on appropriate network elements. This cooperations contributes to a ensure a highly automated network that is also robust, reliable and secure.

The advantage of the proposed mechanism is that the upstream AS can provide protection to the downstream AS from bandwidth-saturating DDoS traffic. The proposed mechanism can also be coupled with policies to trigger how requests are issued. Nevertheless, it is out of scope of this document to elaborate on an exhaustive list of such policies.

How a server determines which network elements should be modified to install appropriate filtering rules is out of scope. A variety of mechanisms and protocols (including NETCONF) may be considered to

exchange information through a communication interface between the server and these underlying elements; the selection of appropriate mechanisms and protocols to be invoked for that interfaces is deployment-specific.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Solution Overview

Network applications have finite resources like CPU cycles, number of processes or threads they can create and use, maximum number of simultaneous connections it can handle, limited resources of the control plane, etc. When processing network traffic, such an application uses these resources to offer its intended task in the most efficient fashion. However, an attacker may be able to prevent the application from performing its intended task by causing the application to exhaust the finite supply of a specific resource.

The complexity and the multitude of potential targets result in making DDoS detection a distributed system over a network. Flood attacks can be detected at the entrance of the network, SYN floods may be detected by firewalls associated to behavioral analysis. Attacks on the link are carried out by sending enough traffic such that the link becomes excessively congested, and legitimate traffic suffers high packet loss. Other possible DDoS attacks are discussed in [RFC4732].

In each of the cases described above, if a network resource detects a potential DDoS attack from a set of IP addresses, the network resource informs its servicing router of all suspect IP addresses that need to be blocked or black-listed for further investigation. That router in-turn propagates the black-listed IP addresses to the access network and the access network blocks traffic from these IP addresses to the customer network thus reducing the effectiveness of the attack. The network resource, after certain duration, requests the rules to block traffic from these IP addresses be removed.

If a blacklisted IPv4 address is shared by multiple subscribers then the side effect of applying the black-list rule will be that traffic from non-attackers will also be blocked by the access network.

4. Protocol Requirements

The protocol requirements for co-operative DDoS mitigation are the following:

- Acknowledgement for the processing of a filtering request and the enforcement of associated countermeasures.
- Mechanism to delete a configured rule.
- Mechanism to convey lifetime of a rule.
- Mechanism to extend the validity of a rule.
- Mechanism to retrieve a list of filtering rules.
- Protocol needs to support "forward compatibility" where the network resource can tell the network entity what version it supports and vice-versa. Any protocol describing attack

mitigations needs forwards compatibility so that new attacks can be described while still allowing older peers (who do not yet understand the new attack) to provide some mitigation.
- The mechanism should support the ability to send a request to multiple destinations (e.g., multi-homing cases).
- Because multiple clients may be allowed to send requests on behalf of a downstream node, the mechanism should allow to signal conflicting requests.
- The request to install a filter may indicate an action (e.g., block, add to a watch list, etc.).
- The mechanism must be transported over a reliable transport.

The security requirements for co-operative DDoS mitigation are the following:

- There must be a mechanism for mutual authentication between the network resource that is signaling black-list rules and the network entity that uses the rules either to propagate the rules upstream or enforces the rules locally to block traffic from attackers.
- Integrity protection is necessary to ensure that a man-in-the-middle (MITM) device does not alter the rules.

o Replay protection is required to ensure that passive attacker does not replay old rules.

5. Protocols for Consideration

An access network can advertise support for filtering rules based on REST APIs. A CPE router should use RESTful APIs discussed in this section to inform the access network of any desired IP filtering rules. If the access network does not advertise support for REST, BGP can be used. The means by which an access network can make this advertisement is outside the scope of this document.

5.1. REST

A network resource could use HTTP to provision and manage filters on the access network. The network resource authenticates itself to the CPE router, which in turn authenticates itself to a server in the access network, creating a two-link chain of transitive authentication between the network resource and the access network. The CPE router validates if the network resource is authorized to signal the black-list rules. Likewise, the server in the access network validates if the CPE router is authorized to signal the black-list rules. To create or purge filters, the network resource sends HTTP requests to the CPE router. The CPE router acts as HTTP proxy, validates the rules and proxies the HTTP requests containing the black-listed IP addresses to the HTTP server in the access

network. When the HTTP proxy receives the associated HTTP response from the HTTP server, it propagates the response back to the network resource.

If an attack is detected by the CPE router then it can act as a HTTP client and signal the black-list rules to the access network. Thus the CPE router plays the role of both HTTP client and HTTP proxy.

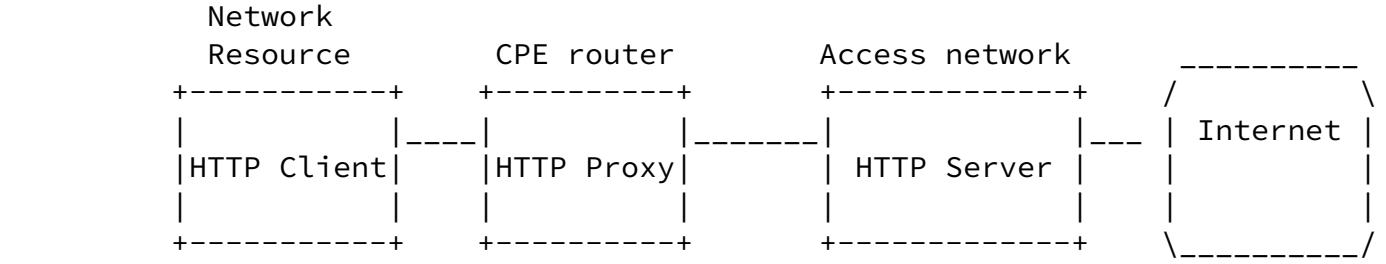

Figure 1

JSON [RFC7159] payloads can be used to convey both filtering rules as well as protocol-specific payload messages that convey request parameters and response information such as errors.

5.1.1. Install black-list rules

An HTTP POST request will be used to push black-list rules to the access network.

```
POST {scheme}://{host}:{port}/.well-known/{version}/{URI suffix}
Accept: application/json
Content-type: application/json
{
   "policy-id": number,
   "traffic-protocol": string,
   "source-protocol-port": string,
   "destination-protocol-port": string,
   "destination-ip": string,
   "source-ip": string,
   "lifetime": number,
   "traffic-rate" : number,
 }
```

Figure 2: POST to install black-list rules

The header fields are described below.

policy-id: Identifier of the policy represented using a number. This identifier must be unique for each policy bound to the same downstream network. This identifier must be generated by the

client and used as an opaque value by the server. This document does not make any assumption about how this identifier is generated.

traffic-protocol: Valid protocol values include tcp and udp.

source-protocol-port: For TCP or UDP: the source range of ports (e.g., 1024-65535).

destination-protocol-port: For TCP or UDP: the destination range of ports (e.g., 443-443). This information is useful to avoid disturbing a group of customers when address sharing is in use [RFC6269].

destination-ip: The destination IP addresses or prefixes.

source-ip: The source IP addresses or prefixes.

lifetime: Lifetime of the policy in seconds. Indicates the validity of a rule. Upon the expiry of this lifetime, and if the request is not reiterated, the rule will be withdrawn at the upstream network. A null value is not allowed.

traffic-rate: This field carries the rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of '0' should result on all traffic for the particular flow to be discarded.

The relative order of two rules is determined by comparing their respective policy identifiers. The rule with lower numeric policy identifier value has higher precedence (and thus will match before) than the rule with higher numeric policy identifier value.

Note: administrative-related clauses may be included as part of the request (such a contract Identifier or a customer identifier). Those clauses are out of scope of this document.

The following example shows POST request to block traffic from attacker IPv6 prefix 2001:db8:abcd:3f01::/64 to network resource using IPv6 address 2002:db8:6401::1 to provide HTTPS web service.

POST https://www.example.com/.well-known/v1/acl

```
Accept: application/json
Content-type: application/json
 {
   "policy-id": 123321333242,
   "traffic-protocol": "tcp",
   "source-protocol-port": "1-65535",
   "destination-protocol-port": "443",
   "destination-ip": "2001:db8:abcd:3f01::/64",
   "source-ip": "2002:db8:6401::1",
   "lifetime": 1800,
   "traffic-rate": 0,
 }
```

Figure 3: POST to install black-list rules

5.1.2. Remove black-list rules

An HTTP DELETE request will be used to delete the black-list rules programmed on the access network.

```
DELETE {scheme}://{host}:{port}/.well-known/{URI suffix}
Accept: application/json
Content-type: application/json
 {
   "policy-id": number
 }
```

Figure 4: DELETE to remove the rules

5.1.3. Retrieving the black-list rules installed

An HTTP GET request will be used to retrieve the black-list rules programmed on the access network.

1) To retrieve all the black-lists rules programmed by the CPE router.

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix}
```

2) To retrieve specific black-list rules programmed by the CPE router.

```
GET {scheme}://{host}:{port}/.well-known/{URI suffix}
Accept: application/json
Content-type: application/json
 {
   "policy-id": number
 }
```

Figure 5: GET to retrieve the rules

5.1.4. TBD

TBD

1. A CPE router can optionally convey metadata describing the attack type and characteristics of the attack to the access network. In some cases, especially with new forms of attack that don't fit existing mitigation mechanisms or exceed network or mitigation capacity, the attack can't be slowed or stopped. The access network might be able to signal its inability to stop the attack (if it is aware) or might be unaware that the attack continues to flow. In such cases where the attack continues, even after filters are requested and installed, the CPE may still need to obtain DDoS mitigation from an external service, outside the scope of this document.

2. The network resource periodically queries the CPE router to check the counters mitigating the attack and the query is recursively propagated upstream till it reaches the access network that has blocked the attack. If the network resource receives response that the counters have not incremented then it can instruct the black-list rules to be removed.

5.2. BGP

BGP defines a mechanism as described in [RFC5575] that can be used to automate inter-domain coordination of traffic filtering, such as what is required in order to mitigate DDoS attacks. However, support for BGP in an access network does not guarantee that traffic filtering will always be honored. Since a CPE router will not receive an acknowledgment for the filtering request, the CPE router should

monitor and apply similar rules in its own network in cases where the upstream network is unable to enforce the filtering rules. In

addition, enforcement of filtering rules of BGP on Internet routers are usually governed by the maximum number of data elements the routers can hold as well as the number of events they are able to process in a given unit of time.

6. IANA Considerations

TODO

7. Security Considerations

If REST is used then HTTPS must be used for data integrity and replay protection. TLS based on client certificate or HTTP authentication must be used to authenticate the network resource signaling the black-list rules.

Special care should be taken in order to ensure that the activation of the proposed mechanism won't have an impact on the stability of the network (including connectivity and services delivered over that network).

Involved functional elements in the cooperation system must establish exchange instructions and notification over a secure and authenticated channel. Adequate filters can be enforced to avoid that nodes outside a trusted domain can inject request such as deleting filtering rules. Nevertheless, attacks can be initiated from within the trusted domain if an entity has been corrupted. Adequate means to monitor trusted nodes should also be enabled.

8. Acknowledgements

Thanks to C. Jacquenet for the discussion and comments.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[REPORT] "Worldwide Infrastructure Security Report", 2014, <http://pages.arbornetworks.com/rs/arbor/images/WISR2014.pdf>.

[RFC4732] Handley, M., Rescorla, E., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

[RFC5575] Marques, P., Sheth, N., Raszuk, R., Greene, B., Mauch, J., and D. McPherson, "Dissemination of Flow Specification Rules", RFC 5575, August 2009.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com

Mike Geller
Cisco Systems, Inc.
3250
Florida 33309
USA

Email: mgeller@cisco.com

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com