

RFC Beautification Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2016

D. Migault
A. Ranjbar
Ericsson
June 28, 2016

Collaboration Agreement for Security Service Function
draft-mglt-i2nsf-ssf-collaboration-00.txt

Abstract

This document specifies a collaboration agreement protocol. The collaboration agreement makes possible individual security services functions (SSF) to collaborate with each other. The collaboration is mostly intended for SSF located in different administrative domains, in which case the collaboration cannot be performed by a shared orchestrator.

The collaboration between SSF in different domains assumes the traffic is steered through the two domains.

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 30, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Introduction	2
3. Collaboration Agreement	3
4. Collaboration Agreement Protocol	4
5. Collaboration Agreement Management operations	6
6. Error Message handling	6
7. Payload Format	7
7.1. Collaboration Agreement Objects	7
7.2. Collaboration Agreement Protocol	11
7.2.1. Collaboration Agreement Protocol Request	11
7.2.2. Collaboration Agreement Protocol Response	12
7.3. Collaboration Agreement Protocol Additional Operations	12
8. Security Considerations	13
9. IANA Considerations	13
10. Acknowledgements	13
11. Normative References	13
Authors' Addresses	13

1. Requirements notation

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].

2. Introduction

Security Service Function (SSF) has been deployed to mitigate and detect malicious traffic and security threats in networks.

A typical use case would consider today's cloud-based services where a data flow is forwarded from the Internet Service Provider to the cloud which hosts the destination service or any on-path services. The services deployed in the cloud are at least partly implemented using a combination of one or more SSF. Similarly, the ISP may also implement a set of on path SSF. The purpose of the collaboration is to enable a SSF running in the cloud administrative domain to take advantage of specific SSF running in the ISP administrative domain. The SSF may be of same type or of different type.

As the SSFs belong to different administrative domains, collaboration between these two SSFs is unlikely to happen through a common shared

orchestrator. To enable the collaboration between individual SSFs, a collaboration agreement protocol is proposed in this document. This protocol is expected to provide: better detection by exchanging real-time information about the detected attacks between SSFs, better mitigation by enforcing mitigation strategies on more effective network segments (e.g. cloud vs ISP), and better resource usage by eliminating the need for frequent deployment of similar service functions and by spreading the tasks among different SSFs.

3. Collaboration Agreement

The SSFs initiating and accepting the collaboration are called, respectively, 'initiator' and 'provider'. The initiator sends to potential providers a Collaboration Agreement (CA), which defines the necessary attributes involved in the collaboration. Such attributes are expected not to be SSF specific. However attributes that characterize the SSF, such as the SFF type, input and output flows, may be part of the CA simply to allow collaborators to define whether or not they are eligible to provide the corresponding services.

For management purposes, the collaboration agreement should also include an 'agreement ID' and a 'duration' indicating its lifespan. It is the responsibility of the 'initiator' to renew the agreement before it expires, although the 'provider' should also be able to notify the former that the agreement needs to be revised or interrupted earlier due to some unexpected event.

Two collaboration modes are envisioned:

- 1) 'Resilient', in which the provider is expected to handle the whole load of that traffic; and
- 2) 'Best Effort', which indicates that the provider supplies the service for a fraction of the load. When the Best Effort mode is chosen, an 'alternate path' indicates where non-treated traffic is forwarded, and 'resource' indicates the resources allocated for the service. The use of 'alternate path' enhances the collaboration between SSFs by allowing the provider to temporarily assign specific amount of resources for handling the packets and send the non-treated traffic through the alternate path to be processed by the initiator. The resources assigned in the Best Effort mode can be expressed in specific ways, such as a combination of various computational resources e.g., CPU, I/O, bandwidth, packet rate, or maximum latency. The manner by which such resources are controlled is left for the provider's implementation (e.g., by leveraging containers and micro services technologies).

Here are the parameters associated to the collaboration agreement:

- o `ca_id`: identifies the collaboration agreement and it can be used later to refer to a specific collaboration agreement.
- o `initiator`: designates the locators (e.g. IP address or FQDN) as well as the authentication credentials associated to the initiator.
- o `provider`: see initiator
- o `collaboration type`: indicates the type of collaboration (Best Effort and Resilient).
- o `resource`: designates the resources agreed on between the initiator and the provider. Note that this parameter is optional as resources are only negotiated when collaboration is in a best effort mode.
- o `SSF type`: designates the type of the security instances running.
- o `expiration time`: designates the expiration of the collaboration.
- o `interconnections`: defines how the interconnection between the initiator and the provider is performed. This includes the definition of the alternate path.
- o `direction`: defines if the provider is expected to be in front of the initiator or behind it.

4. Collaboration Agreement Protocol

The purpose of the collaboration agreement protocol is to negotiate between the initiator and provider and make an agreement for cooperation between SSFs placed in a single or multiple domains. Currently, the collaboration agreement protocol is always originated from the initiator. In other words, the provider is not initiating the exchanges as to announce what it can provide.

The collaboration agreement protocol should include the following attributes:

- o `ca_id`: this is the collaboration agreement identifier. In a case the value is not acceptable, an `ERROR_UNACCEPTABLE_CA_ID` MUST be returned. There are, however, little reasons such a collision occurs. If such a collision occurs, the negotiation is aborted and must be restarted with a new `ca_id`.

- o initiator: it includes information that the initiator offers to the provider. Upon receiving the request for collaboration, the provider may reject the collaboration agreement by sending a `ERROR_UNACCEPTABLE_INITIATOR`.
- o provider: it consists of information about the provider to be verified by the initiator. Upon receiving this information, the initiator may abort the negotiation with `ERROR_UNACCEPTABLE_PROVIDER`. The reason for refusing the provider, may be that the provider is not in a white list or that the provider has been explicitly banned by the initiator.
- o resource: it represents allocated resources by the provider for collaboration with the initiator. When the collaboration mode is set to Resilient, the resource is not expected to be provided by the provider. For the best effort mode, the resource provided by the provider may consider the indication provided by the initiator or not. Given the resource provided by the provider, the initiator is likely to close the collaboration or to accept it.

In order to define a flexible framework, the negotiation steps between the initiator and provider is designed as mentioned below:

1. The initiator provides a list of proposals to the provider
2. A proposal may contain multiple proposition for a given attribute. For example, let P1 be a proposal offered by the initiator. In this case, the initiator may be willing to make an agreement with the providers either in Best Effort or Resilient modes. In this case, the initiator will set P1 with an object of collaboration type set to Best Effort AND an object of collaboration type set to Resilient.
3. When multiple proposals are received by the provider, the provider is expected to choose a single proposal. The chosen proposal is the one that contains the attributes that fits the provider.
4. When a proposal is chosen, the provider must select for each attribute the preferred value. More especially, when multiple values for a same attribute type are available, the provider selects the preferred value for that attribute. Also, the chosen proposal must have the same amount of attribute types which means the provider is not allowed to remove some attributes or selectively reject attributes.

5. The provider may send an acceptable proposal to the initiator. If none of the proposals are acceptable by the provider, the provider returns a `ERROR_UNACCEPTABLE_PROPOSALS`.

5. Collaboration Agreement Management operations

Once the collaboration agreement has been agreed between the initiator and provider, the following actions need to be considered during its life cycle.

- o `END_AGREEMENT`: This action can be performed by either peers, that is to say the initiator or the provider. This action requires the `ca_id` and credentials to identify peers in the agreement.
- o `UPDATE_EXPIRATION_DATE`: This action is initiated by either peers. It intends to update the expiration date. The expiration date can be extended or advanced. The input parameters are the `ca_id` and the new expiration date. The possible responses are to accept or reject this request. In case of rejection, `ERROR_UNACCEPTABLE_NEW_EXPIRATION_DATE` is sent to the requested peer.
- o `UPDATE_RESOURCE`: This action is expected to be triggered by the provider. It indicates the amount of resources the provider offers for collaboration. This is an informative message. It may be useful for the initiator to know how much resource will be dedicated to the collaboration by the provider so it can adjust its strategy.
- o `REDIRECT_SSF`: This action is triggered when peers change their location. This action is initiated by either peers. It may result in changes in Alternate Path in case of Best Effort mode.

6. Error Message handling

The following Error message have been considered so far:

```
ERROR_UNACCEPTABLE_CA_ID
ERROR_UNACCEPTABLE_PROVIDER
ERROR_UNACCEPTABLE_INITIATOR
ERROR_UNACCEPTABLE_PROPOSALS
ERROR_UNACCEPTABLE_NEW_EXPIRATION_DATE
```

7. Payload Format

7.1. Collaboration Agreement Objects

This section represents the Collaboration Agreement object. The collaboration agreement is an object with properties. Some of these properties are object themselves. In order to enrich the object definition, the Collaboration is defined on different objects including 'peer' and 'resource' objects.

'Peer' object represents the necessary information associated to a peer. A peer can be either the initiator or provider. The description of a peer object is as follows:

```
{
  "peer": {
    "type": "object",
    "description": "provides different elements associated to the
                    initiator or the provider. This includes
                    location as well as authentication credentials",
    "properties": {
      "rsakey": {
        "type": "string",
        "description": "RSA public key used to identify the
                        initiator"
      },
      "cert": {
        "type": "array",
        "description": "list of certificates to authenticate the
                        initiator"
      },
      "fqdn": {
        "type": "string",
        "description": " FQDN associated to the initiator"
      },
      "ipv4": {
        "type": "string",
        "description": "IPv4 address used to reach the initiator"
      },
      "ipv6": {
        "type": "string",
        "description": "IPv6 address used to reach the initiator"
      }
    }
  }
}
```

The following object designates the resources agreed between the initiator and the provider.

```
{
  "resource": {
    "type": "object",
    "description": "resource engaged into the collaboration",
    "properties": {
      "cpu": {
        "type": "number",
        "description": "cpu limit"
      },
      "memory": {
        "type": "number",
        "description": "memory limit"
      },
      "net": {
        "type": "number",
        "description": "net limit"
      },
      "blkio": {
        "type": "number",
        "description": "block limit"
      }
    }
  }
}
```

The collaboration type is defined as follow:

TYPE	CODE
Resilient	0
Best Effort	1

SSF instance types can be extended to any number of available services. We do not limit SSF types and we expect to extend this number in future. Some example SSFs can be defined as follows:

TYPE	CODE
Rate limiting	0
DNSoverTCP	1
PacketDropper	2

The following object is a Collaboration Agreement object which includes several properties to define an agreement between the provider and initiator.

```
{
  "type": "Collaboration Agreement",
  "description": "This object designates the Collaboration
                  Agreement properties",
  "properties": {
    "ca_id" : {
      "type": "number",
      "description" : "unique identifier of the Collaboration
                      agreement"
    },
    "initiator": {
      "type": "peer",
      "description": "provides the different elements associated
                      to the initiator. This includes location
                      as well as authentication credentials"
    },
    "provider": {
      "type": "peer",
      "description": "provides the different elements associated
                      to the provider. This includes location as
                      well as authentication credentials"
    },
    "collaboration_type": {
      "type": "number",
      "description": "defines whether the type of the
                      collaboration"
    },
    "security_service_instance_type": {
      "type": "number",
      "description": "the type of security service instance"
    },
    "interconnections":{
      "type": "interconnections",
      "description": "the type of security service instance"
    },
    "resource":{
      "type": "resource",
      "description": "the type of security service instance"
    },
    "direction":{
      "type": "direction",
      "description": "indicates whether the provider MUST
                      be placed downstream or upstream"
    }
  }
}
```

7.2. Collaboration Agreement Protocol

The collaboration agreement protocol can be defined as request or response objects.

7.2.1. Collaboration Agreement Protocol Request

The following object defines the request object which includes information about initiator, resources and a set of proposal objects.

```
{
  "type": "collaboration protocol agreement request",
  "description": "object",
  "properties": {
    "ca_id" : {
      "type": "number",
      "description" : "unique identifier for collaboration
                      agreement"
    },
    "initiator":{
      "type": "peer",
      "description": "provides different elements associated
                      to the initiator. This includes location
                      as well as authentication credentials"
    },
    "informative resource requested":{
      "type": "resource",
      "description": "the type of security service instance"
    },
    "proposals":{
      "type": "Array",
      "description": "Array of proposals offered by the initiator"
    }
  }
}
```

A proposal object can also be defined as follows:

```
{
  "type": "object",
  "description": "A single proposal with a set of attributes.
                  The expected attribute types are collaboration
                  type, security service instance type and
                  interconnections",
  "properties": {
    "proposal_id": {
      "type": "number"
    }
    "proposed-attribute": {
      "type": "object"
      "properties": {
        "attribute-type": {
          type: string
        }
        "attribute-values": {
          "type": "array"
          "items": {
            attribute-value
          }
        }
      }
    }
  }
}
```

7.2.2. Collaboration Agreement Protocol Response

The response object is similar to the request object except that:

- o The response must include a provider object.
- o The proposed list of attribute values must be of size one with the chosen value.

7.3. Collaboration Agreement Protocol Additional Operations

When the initiator and provider are placed in different domains, additional orchestration operations might be needed between domains to make an agreement. Moreover, in case of Best Effort mode, additional operations is needed to establish an alternate path and separate the treated traffic from non-treated traffic e.g. by deploying classifiers on the path.

8. Security Considerations

9. IANA Considerations

10. Acknowledgements

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Alireza Ranjbar
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Phone: +358-442992904
Email: alireza.ranjbar@ericsson.com