Workgroup: Network Working Group
Internet-Draft: draft-xu-savax-control-02
Published: 20 May 2022
Intended Status: Informational
Expires: 21 November 2022
Authors: K. Xu
Tsinghua University
J. Wu
Tsinghua University
X. Wang
Tsinghua University
Y. Guo
Tsinghua University

Control Plane of Inter-Domain Source Address Validation Architecture

Abstract

Because the Internet forwards packets according to the IP destination address, packet forwarding typically takes place without inspection of the source address and malicious attacks have been launched using spoofed source addresses. The inter-domain source address validation architecture is an effort to enhance the Internet by using state machine to generate consistent tags. When communicating between two end hosts at different ADs of IPv6 network, tags will be added to the packets to identify the authenticity of the IPv6 source address.

This memo focus on the control plane of the SAVA-X mechanism.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Inter-Domain Source Address Validation (SAVA-X) mechanism establishes a trust alliance among Address Domains (AD), maintains a one-to-one state machine among ADs with AD Control Server (ACS), generates a consistent tag, and deploys the tag to the ADs' border router (AER). The AER of the source AD adds a tag to identify the identity of the AD to the packet originating from one AD and sinking in another AD. The AER of the destination AD verifies the source address by validating the correctness of the tag to determine whether it is a packet with a forged source address.

In the process of packet forwarding, if the source address and the destination address of this packet both are addresses in the trust alliance, however the tag is not added or incorrectly added, AER of the destination AD determines that the source address is forged and directly discards this packet. The destination AD forwards the packet directly for packets whose source address is an address outside the trust alliance.

This document mainly studies the relevant specifications of the control plane of the inter-domain source address validation architecture mechanism between ADs, which will protect IPv6 networks from being forged source address. You could see [RFC8200] for more details about IPv6. It stipulates the design of the consortium blockchain, the nodes' joining and exiting, the maintenance of trust alliance information based on the consortium blockchain, and the maintenance of the state machine. Its promotion and application can realize the standardization of the control plane in the SAVA-X to facilitate the related equipment developed by different manufacturers and organizations to cooperate to accomplish the inter-domain source address validation jointly.

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 RFC 2119, BCP 14 [RFC2119] and indicate requirement levels for compliant CoAP implementations.

2. Terminology and Abbreviation

Abbreviation	Description
AD	Address Domain, the unit of a trust alliance, which is an address set consisting of all IPv6 addresses corresponding to an IPv6 address prefix.
TA	Trust Alliance, the IPv6 network that uses the SAVA-X mechanism.
STA	sub-Trust Alliance, parts of TA.
STA-admin	STA Administrator, the administrator of STA.
ACS	AD Control Server, the server that matains state machine with other ACS and distribute information to AER.
AER	AD border router, which is placed at the boundary of an AD of STA.
ADID	The identity of an AD.
ADID_Rec	The record of a number of an AD.
ARI_Rec	The record with relavent information of an AD or STA.
API_Rec	The record of prefix of an AD or STA.
CSR	Certificate Signing Request, which is used for an AD or STA to join or exit the consortium blockchain.
SM	State Machine, which is maintained by a pair of ACS to generate tags.
Tag	The authentic identification of source address of a packet.

Table 1

3. The design of the Consortium Blockchain

As discussed in the introduction, consortium blockchain will be used in SAVA-X mechanism.

3.1. Trust Alliance

Trust Alliance (TA) is a hierarchical structure. Address domains (AD) are assigned into different sub-trust alliances (STA) according to geographic location, economic relationship, political relationship, social relationship, and military relationship. AD is the minimum unit for trust. The one-to-one maintenance state machine between ADs located in the same layer of sub-trust alliance generates consistent tags and deploys the tags to their AERs. The ADs in each sub-trust alliance elect a master AD node. The master AD node represents the sub-trust alliance and maintains the alliance-level state machine with other master AD nodes to generate alliance-level tags. When communicating across sub-trust alliances, it is necessary to achieve the feature of tag replacement.

The AD in the SAVA-X must be located in a specific sub-trust alliance. According to its position in the SAVA-X, AD can be divided into three roles: primary address domain, boundary address domain, and ordinary address domain which is neither primary nor boundary address domain.

* The primary address domain is the representative node of the aforementioned sub-trust alliance and is used to establish connection with the primary address domain of other sub-trust alliances. In this way, the relationship between trust alliances finally forms a tree-like relationship, and there will be no direct relationship between address domains under the same branch.

* The boundary address domain is the address domain located at the boundary of the sub-trust alliance. It sends the packet to other sub-trust alliances or outside the trust alliance.

* The ordinary address domain is neither the primary address domain nor the address domain of the boundary address domain.

Due to the uncontrollable packet forwarding path, in SAVA-X, a virtual address domain needs to be set up as a non-boundary AD to communicate with the sub-trust alliance outside or receive packets sent from outside the sub-trust alliance to maintain the state machine. The virtual AD is recorded as AD_V (Virtual Address Domain). When a packet from an AD in a sub-trust alliance needs to be sent outside the sub-trust alliance, but there are multiple paths to the destination AD in the sub-trust alliance, the sub-trust alliance may have multiple boundary ADs to reach the destination AD.

The sub-trust alliance doesnot know which boundary AD will be selected during the packet forwarding. Therefore, the primary function of AD_V is to prevent this by specifying the specific tag that should be added to the packet sent to the external address domain of the sub-trust alliance.

What's more, the tag needs to be verified by the boundary address domain of the sub-trust alliance. Therefore, the boundary AD also needs to receive the tags maintained by the AD and AD_V in the trust alliance. As a tag for communicating data between the non-primary address domain and the external address domain of the sub-trust alliance.

3.2. Consortium Blockchain

To simplify the control plane's design and avoid the single point failure to subvert the SAVA-X, we design the SAVA-X with a decentralized infrastructure which will store the information of the trust alliance.

The consortium blockchain is composed of the trust alliance management committee chain and several sub-chains. Among them, the management committee chain is composed of several nodes to manage the administrator nodes of each sub-chain. The consortium blockchain records information of the sub-trust alliance administrator node, named as STA-admin (Sub Trust Alliance administrator), and member list of each sub-chain which are packaged and submitted by the STA-admin. Each sub-trust alliance has one STA-admin that is assumed by a specific elected AD in the sub-trust alliance. The AD in the same sub-trust alliance forms a private chain to maintain the information of the members of the sub-trust alliance jointly. The STA-admin in each sub-trust alliance is responsible for managing the joining and exiting of the sub-trust alliance node. The STA-admin of each sub-trust alliance maintains the relationship of the members in each sub-trust alliance through the trust alliance management committee chain.

3.3. Joining and Exiting

3.3.1. Node Joining

This is the admission of joining of the sub-trustalliance member AD. The prerequisite for the AD to join the sub-trust alliance is to have a certificate issued by the STA-admin firstly. AD's Address Control Server (ACS), which will maintain state machine with other ACS and distribute alliance information and other information to AER, submits a Certificate Signing Request file to the STA-admin of the sub-trust alliance that it wants to join to request the certificate. The CSR file includes ADID, ACS address information,

IPv6 address prefix information, and its public key information. If the file is valid, STA-admin verifies the file, generates a node certificate, packages the AD's information into a block, and updates the list of members of the sub-consortium. STA-admin submits the latest block to the consortium blockchain, and the consortium blockchain updates the list of alliance members of the entire trust alliance.

When a sub-trust alliance want to join the trust alliance, STA-admin submits a CSR file to the consortium blockchain, including the member information list in the sub-chain and the information of the STA-admin. It requires offline negotiation and cooperation to apply for joining the consortium blockchain. The consortium blockchain management committee verifies the validity of the request, issues administrator certificates, and updates block information. The STA-admins in the current trust alliance jointly maintain a management committee chain, manage the administrator certificates of each sub-trust alliance, and use the certificates for encrypted communication. STA-admin submits the list of members of the sub-trust alliance to the consortium blockchain and joins the entire trust alliance.

After a node joins the consortium blockchain, there is an Effecting Time and an Expiration Time in the CSR file. STA-admin will assign the sub-trust alliance member with an ADID number if it does not contain the ADID information in the submitted information. The consortium blockchain will give the permitted sub-trust alliance a sub-trust alliance number if the information submitted by the sub-trust alliance does not have the sub-trust alliance number. If there is a conflict between the submitted information and the returned information, the returned ADID or sub-trust alliance number will be selected.

3.3.2. Node Exiting

The member node needs to submit the CSR file again to delete its relevant information. Its STA-admin decides whether to allow it exit or not. After passing the validation, nodes of the sub-blockchain delete the relevant member information. It also needs to submit a CSR file for the exit of the sub-trust alliance node, which the alliance management committee decides whether to allow it. After validating the receiving exit request, other sub-trust alliance administrator nodes need to delete their maintenance-related sub alliance node information.

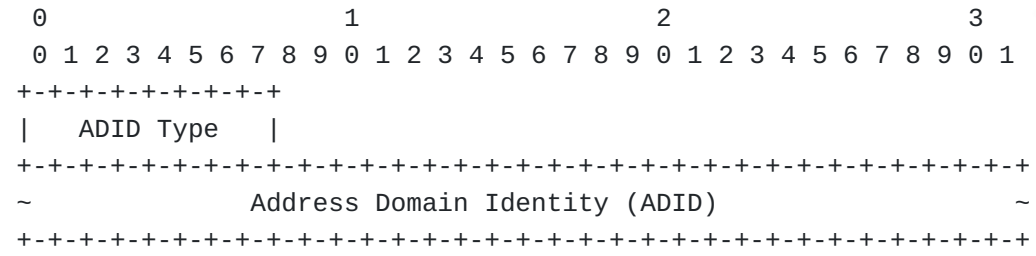
4. Alliance Information and State Machine Maintenance based on the Consortium Blockchain

On the AER of the destination AD, to validate the tag, it is first necessary to find out the sub-trust alliance number from the source address of the arriving packet and find out its corresponding source Address Domain Identity (ADID) number. Then find the currently valid tag according to the ADID number. The generation of the tag requires the maintenance of the state machine between the ACSes. In SAVA-X, member ADs need to inform each other of their sub-trust alliance number, ADID number, AD role, ACS address, and IPv6 address prefix. The members interact with each other with the state machine information according to the hierarchical division structure after obtaining the basic information of the other members. And use the tags generated by the state machine during the packet forwarding after the specified time to add and validate the tags.

The relevant information needs to be stored in the sub-chains, where the node is located after joining the consortium blockchain. The information stored on the consortium blockchain needs the content specified in the following three message formats, namely ADID_Rec, ARI_Rec, and API_Rec. We give the packet format required by SAVA-X in the control plane as follows.

4.1. Address Domain Identity Record

Address Domain Identity Record (ADID_Rec) is used to identify an address domain uniquely in the trust alliacne.



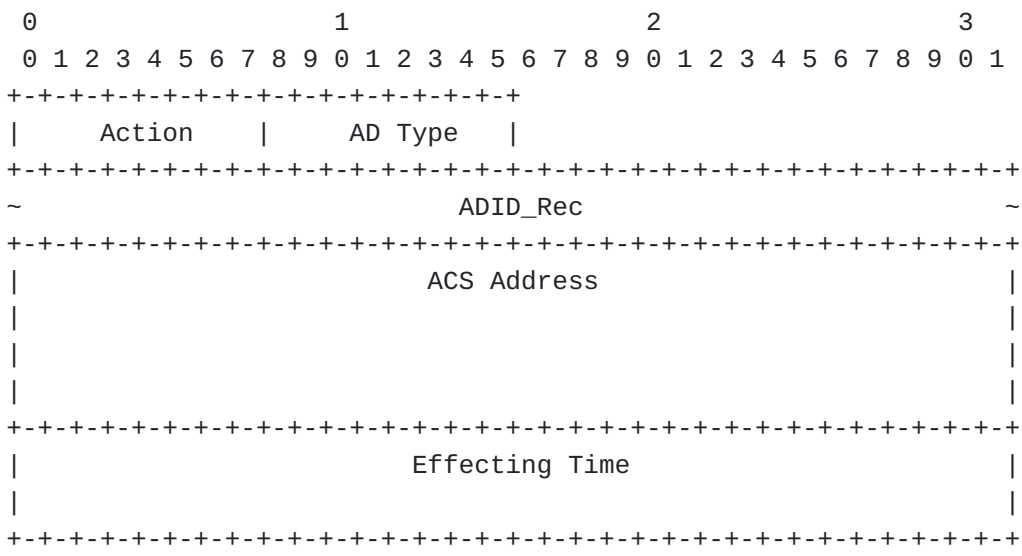
ADID Type: 8-bit Type of ADID, 1 for 16-bits AS number, 2 for 32-bits AS number and other unassigned.

ADID: The 16-bit or 32-bit ADID number. Its value can be the AS number or the number assigned by the consortium blockchain, and the specific length is determined by the ADID Type field. When each bit of ADID is 1, it represents that the AER requests the information of all members from ACS.

4.2. AD Registration Information Record

AD Registration Information Record (ARI_Rec) is the registration information record of AD, which is used to record the necessary

information required to register a specific member AD. The ACS and AD establish connections.



Action: 8-bit instruction to add (ADD=1) or delete (DEL=2) this record. Others are unassigned.

AD Type: 8-bit unsigned number indicating the role of AD. 0 for ordinary AD, 1 for primary AD and 2 for boundary AD. Others are unassigned.

ADID_Rec: Reference the ADID_Rec packet.

ACS Address: 128-bit the IPv6 address of ACS.

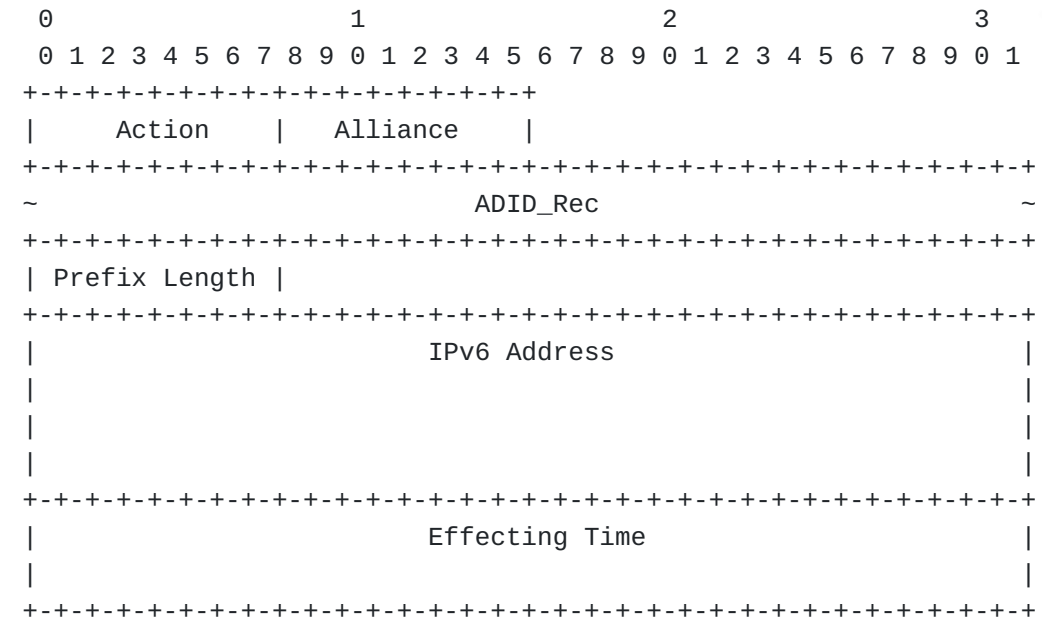
Effecting Time: 64-bit time specifies when this record is applied. It is recommended to use the 64 bits struct timeval format for the effecting time of the execution of this record. If all bits of this field are 0 or earlier than the current time, it means that it takes effect immediately.

The role of address domain is essential, and each address domain needs to be assigned a corresponding role according to its position in the sub-trust alliance. A sub-trust alliance needs to set one (and only one) primary address domain. It serves as the representative of the sub-trust alliance. The tag generated by its ACS and the ACSes of other sub-trust alliances' primary address domain maintains the state machine to generate the tag of the sub-trust alliance, and it issues the tag to the boundary address domain of the sub-trust alliance. A specific recommendation of a consensus algorithm could generate the primary address domain or be directly specified by the operator of the address domain with offline negotiation. The boundary address domain means that packet forwards outside the address domain is no longer in the current sub-trust

alliance. The primary address domain can be a boundary address domain or not. The tag replacement may occur in the boundary address domain. The ordinary address domain is neither a primary address domain nor a boundary address domain.

4.3. AD Prefix Information Record

AD Prefix Information Record (API_Rec) is the prefix information corresponds to the prefix of a specific AD. An AD may have more than one prefix, so registration information and prefix information records must be separated.



Action: 8-bit instruction to add (ADD=1) or delete (DEL=2) this record. Others are unassigned.

Alliance: 8-bit the sub-trust alliance number.

ADID_Rec: Reference the ADID_Rec packet.

Prefix Length: 8-bit the length of the IPv6 prefix.

IPv6 Address: 128-bit indicates a certain address prefix operated by the corresponding member AD using together with Prefix Length.

Effecting Time: 64-bit time specifies when this record is applied. It is recommended to use the 64 bits struct timeval format for the effecting time of the execution of this record. If all bits of this field are 0 or earlier than the current time, it means that it takes effect immediately.

ARI_Rec and API_Rec are required to store the AD information in the consortium blockchain and send it to all AERs of their AD with the communication protocol between ACS and AER.

5. Time Synchronization

Due to the time error between the border routers of the member ADs, to ensure the correct operation of the tag validation, it is necessary to make each device including ACS and AER in the trust alliance calibrate to the same clock reference. The NTP protocol could achieve this goal. You could see [RFC5905] for more details.

Although the NTP protocol can guarantee the time synchronization between AERs, there may inevitably still have a slight time difference between AERs and ACSes. Therefore, each AER sets a shared time slice. With this time slice, both the expired tag and the new tag are considered valid. That is, more than one tag is valid for a while. The destination AD needs to validate all valid tags belonging to a specific source AD. The tag is correct if one of the tags is validated.

Assuming that the maximum time difference between AER and ACS is te, we set a shared time slice with a length of 2te between two adjacent tags. In this shared time slice, the two tags before and after are valid. The validity period of the tag with the shared time slice is shown below, see Figure 1.

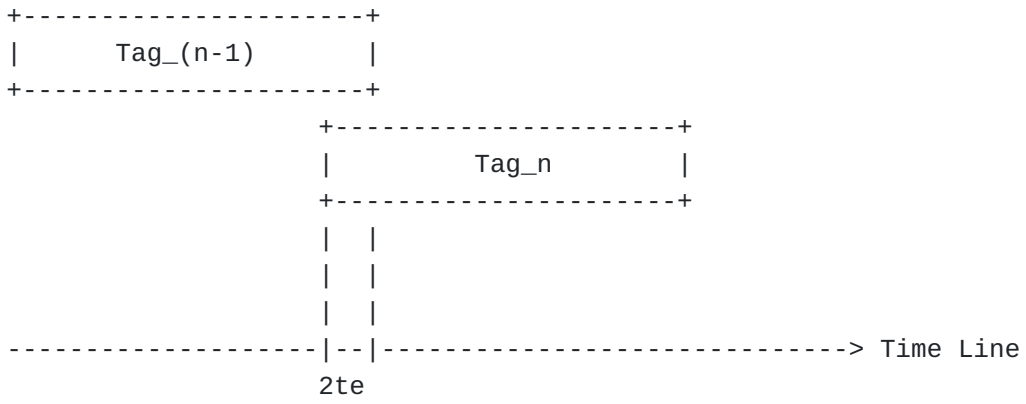


Figure 1: Validity period of tag with the shared time slice

In addition to the time difference, we should also take into account the packet transmission delay in the network. Set the minimum delay to td_min and the maximum delay to td_max. The expiration of Tag_n should be extended td_max later, and the beginning of Tag_(n+1) validity period should be delayed td_min later. The shared time slice and tag validity period corrected according to transmission delay are shown as follows, see Figure 2.

Figure 2: Validity period of tag with the shared time slice after modified

The expiration of the Tag_n is extended to te+td_max, and the beginning of the Tag_(n+1) is extended to te-td_min. The parameters such as te, td_min, td_max, the period of the shared time slice, and tag validity period are determined by the destination based on the actual network environment. Therefore, the lifecyle of a tag is as: lifecycle = Transition Interval + 2te --td_min + td_max.

6. Security Consideration

This present memo doesnot find any security problem.

7. IANA Consideration

This document makes no requests of IANA.

8. Acknowledgements

Much of the content of this document is the expansion of the IETF [RFC5210] in inter-domain level. Thanks to the effort of pioneers.

9. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5210] Wu, J., Bi, J., Li, X., Ren, G., Xu, K., and M. Williams, "A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience", RFC 5210, DOI 10.17487/RFC5210, June 2008, <https://www.rfc-editor.org/info/rfc5210>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms

Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

Authors' Addresses

Ke Xu
Computer Science, Tsinghua University
Qinghuayuan street, Haidian District
Beijing
100084
China

Email: xuke@tsinghua.edu.cn

Jianping Wu
Computer Science, Tsinghua University
Qinghuayuan street, Haidian District
Beijing
100084
China

Email: jianping@cernet.edu.cn

Xiaoliang Wang
Computer Science, Tsinghua University
Qinghuayuan street, Haidian District
Beijing
100084
China

Email: wangxiaoliang0623@foxmail.com

Yangfei Guo
Institute for Network Sciences and Cyberspace, Tsinghua University
Qinghuayuan street, Haidian District
Beijing
100084
China

Email: guoyangf19@mails.tsinghua.edu.cn