May 2022

ACCESS CONTROL POLICY ENFORCEMENT THROUGH ROUTE-BASED MICRO-SEGMENTATION AND CONTRACT TAGS

Victor Moreno

Sanjay Hooda

Steve Wood

ACCESS CONTROL POLICY ENFORCEMENT THROUGH ROUTE-BASED MICRO-SEGMENTATION AND CONTRACT TAGS

AUTHORS:
Victor Moreno
Sanjay Hooda
Steve Wood

ABSTRACT

Large networks service a significant number of endpoints, each of which access a sizable number of applications. This results in the definition of access control policies that are extremely lengthy and therefore difficult to render in common network elements, such as routers and switches, due to the limited amount of memory (such as ternary content-addressable memory (TCAM)) that is included in such platforms for the enforcement of policies. To address the type of challenge that was described above, various solutions are provided herein through several techniques. A first technique supports, among other things, the scaling of the access control entries (ACEs) in a network for specific deployment by converting an ACE to a route control entry. A second technique supports, among other things, the disaggregation of access control policies and the efficient distribution of their enforcement. According to this technique, access control attributes may be evaluated at different network locations to optimize scale by localizing the evaluation of matching criteria to the places in which it requires the minimum amount of state creation and maintenance. A policy may be disaggregated at the orchestrator, its components may be distributed to the different enforcement points, and a tagging mechanism may be used to unify the policy as its elements are dispersed across multiple enforcement points.

DETAILED DESCRIPTION

The enforcement of access control policies on network platforms has traditionally been limited by the capacity of the available memory (such as ternary content-addressable memory (TCAM)) that is included in such platforms for the enforcement of policies. In many cases, the length of the required policies may exceed the capacity of the available TCAM. Consequently, a mechanism to enable the enforcement of very large access control policies on standard routing platforms is desirable.

Further, large networks service a significant number of endpoints, each of which access a sizable number of applications. This results in the definition of access control policies that are extremely long and therefore difficult to render in common network elements (such as routers and switches) due to the limited capacity of hardware tables. One approach that is employed by known solutions to such a challenge encompasses a grouping of endpoints as a means of reducing and simplifying access control policies. Such an approach classifies source endpoints into groups at the ingress of the network, includes security or scalable group tags (SGTs) in the packet headers to reflect a group classification, and defers the full enforcement of a policy to the egress of the network.

Although the logic that is employed in such an approach helps to alleviate the scale problem, hot spots of large data structures in memory may still form in the network. This is particularly true for systems in which the enforcement points are at points of traffic aggregation in which the connectivity for many endpoints converges. Furthermore, there are operations (in, for example, defense networks) in which a grouping of endpoints is not an accepted practice but the rendering of granular policies on a per-prefix basis is required. There are scalability improvements that may be pursued by disaggregating, restructuring, and distributing the classification criteria matching process and the enforcement of associated policy actions.

To address the types of challenges that were described above, various solutions are provided herein through several techniques. Each of the techniques will be briefly introduced below and then described in detail later in the instant narrative.

A first technique supports, among other things, a method for the enforcement of very large access control policies that does not depend upon the TCAM capacity that is available in a platform. Rather, the technique leverages the forwarding table capacity of the platform which is a much larger memory table. Consequently, this technique leverages the possibility of enforcing an access control policy in the routing plane in order to avoid the scale limitations of the TCAM-based access control enforcement tables that are normally available in a router.

Certain enterprise entities are pursuing a native zero-trust-like environment within their own private networks. One example of such a deployment can be found in networks that are built for national defense departments in which a central information technology

(IT) provider services a multitude of independent federal agencies and other operations over which they have no control. In such an environment, it is required that access control policies be enforced at ingress and at the edges of the IT provider's network in order to secure the flow of traffic at the closest point of control that is managed by the central IT provider.

Each agency that is connected to the central IT provider will have a large number of endpoints (in one particular example, more than 15,000), with each endpoint being capable of accessing a relatively rich mix of applications (e.g., on average 80 or more on each endpoint). For operational and compliance reasons, it is required that access control policies be defined on a per-endpoint basis and that access control policies follow a whitelist model in which only the traffic that is explicitly permitted is allowed to enter the network.

The net effect of above requirements is that the edge routers and the devices of the central IT provider's network need to be able to classify traffic for thousands of endpoints and they need to accommodate access control entries in excess of one million entries resulting in a very granular micro-segmented environment.

Aspects of this technique address the latter of the above-described challenges – i.e., the need to enforce millions of policy entries on common routing platforms. While route control to provide access control has been used for macro-segmentation with virtual routing and forwarding (VRF) solutions and virtual private networks (VPNs), aspects of the presented techniques support a mechanism for delivering micro-segmentation within such macro-segments (e.g., VRFs) while still leveraging route control in smaller forwarding contexts within the VRFs.

A second technique, as referenced above, supports, among other things, a method for the disaggregation of the policy enforcement, thus delivering improved scalability. Aspects of this technique disaggregate a policy and separate the evaluation of endpoints from the evaluation of traffic types. In contrast, some existing network access provisioning and management solutions evaluate these criteria jointly at egress, restricting the ability to distribute enforcement and scale. More importantly, aspects of the presented technique do not have a hard dependency on the grouping of endpoints, which allows for the deployment of large-scale policies that are articulated in terms of Internet Protocol (IP) addresses

(rather than source groups) in a distributed manner without requiring larger memory tables on the network devices.

Turning to the first technique, as referenced above, aspects of this technique support, among other things, assigning to each user a dedicated forwarding context or table, steering all of the traffic from that user through its dedicated forwarding context (e.g., micro-segment), and constraining the routes that may be installed on that table to only the routes for the applications that the specific user is allowed to access. Aspects of the presented technique supports a method by which micro-segments (forwarding contexts) are instantiated, the way in which micro-segments are efficiently populated with routes that are reflective of a policy, the relationship of such micro-segments to their parent VRF, and the mechanisms by which traffic may be multiplexed and demultiplexed into and out of the different micro-segments.

Within the presented technique resides the concept of constraining the routes that are available to each user, rather than programming a comprehensive access control list (ACL) on an edge router, to convert an access control policy into a route policy. At a high level, the presented technique encompasses assigning a dedicated forwarding context to each user so that all of the traffic to and from that user is handled through the user's dedicated forwarding table and constraining the routes that are present in each per-user forwarding table (VRF) to only those routes for the specific applications that a user is allowed to access.

According to aspects of this technique, each endpoint is assigned its own forwarding context and the information that is contained in each forwarding context is limited to only the destinations that the endpoint is allowed to reach. Thus, an endpoint will have its reachability constrained to only those destinations that are available in its forwarding context. Whether a destination is permitted or not is defined by the access control policy for the endpoint. This yields a level of forwarding segmentation that is much more granular than that which is achieved through VRFs and the micro-segments exist within the coarser structure of a VRF (or a virtual local area network (VLAN) in a Layer 2 (L2) implementation).

The access control policy that is enforced follows a whitelist model in which only those communications which are explicitly allowed are permitted. In order to achieve this

level of micro-segmentation, several mechanisms must be implemented (according to aspects of the first technique as described below) beyond the general notion of forwarding-based micro-segmentation, where the below-described mechanisms enhance the forwarding path and the population of routes in order to achieve the proposed micro-segmentation based on forwarding information.

Aspects of the first technique may be further explicated through six functional blocks, each of which will be described below.

A first functional block encompasses micro-context creation and assignment logic. For each endpoint IP address that is discovered at the boundary router for a system, a forwarding micro-context may be created as an indexed section of the forwarding table. The index may be derived through a simple heuristic or by a more sophisticated mapping function or service. As just one example, the index may be the actual IP address of the endpoint for which the micro-context is created, resulting in a simple one-to-one mapping between endpoints and micro-contexts. Under a more involved example, endpoints may be grouped and the endpoints in a group may share a micro-context. In such a case, the index may be a group number and the endpoint assignment may be a mapping function between the list of endpoint IP addresses and the group to which they belong (and the group's index). The micro-contexts may be nested within a VRF and may be used for the enforcement of policy, but they do not replace the VRF for the purposes of forwarding. Thus, VPNs are not established amongst micro-contexts on different routers as they would be established amongst VRFs on different routers.

A second functional block encompasses the translation of an access control policy into contexts and routes and the population of micro-contexts. Access control policies may be articulated in terms of the involved endpoints (e.g., a source and a destination), the type of traffic that flows between the endpoints (e.g., a classifier), and the action that is to be taken (e.g., permit, deny, or other). In order to take an action, there must first be a match on the tuple of source and destination endpoint addresses and classifier information. According to aspects of this technique, a source address match is achieved through the demultiplexing mechanisms (that are explained below in connection with a third functional block) in which all of the traffic originating from a particular source may be steered to a specific micro-context. A destination address match is achieved through a destination

lookup within the micro-context. A classifier match may also be achieved through the demultiplexing mechanisms (as explained below). A deny action may be implemented by excluding a destination from the micro-context of a source. A permit action may be implemented by including a destination within the micro-context of a source. A redirect action may be implemented by altering the next hop for the destination that was found within the micro-context (i.e., this is an override of the forwarding action that is described below in connection with a fourth functional block).

A policy may be maintained in the control plane of a network, which is preferably a demand-based control plane such as that which is found in the Location Identifier Separation Protocol (LISP). The actions of permit and deny may be associated with the response that the mapping system provides to a query at lookup time, thus the inclusion or exclusion of a route in a micro-context in order to implement an action is enforced by the control plane at the time of responding to a query for the resolution of a destination (in the context of a particular source).

A third functional block encompasses the demultiplexing of clear IP external traffic into micro-contexts (based on a source IP address (i.e., an ingress point)). For example, the boundary router to a system will receive clear IP traffic (i.e., traffic without specific metadata) from external networks and will need to steer the traffic that originated from different endpoints into their corresponding micro-context. In order to achieve this, the source IP address of the packet may be treated as a label that indicates to the receiving router which micro-context should be used to evaluate the action that is to be taken on the packet. The concept is similar to how an Institute of Electrical and Electronics Engineers (IEEE) 802.1Q standard (dot1Q) tag is used to associate traffic in a trunk port to a particular VLAN. However, under aspects of the first technique the existing source IP address is employed rather than a separate tag. The implication is that any IP system can connect to this type of interface without requiring any special configuration to achieve the demultiplexing. Since the interface that is facing the external network is a Layer 3 (L3) interface, it will have an IP address. This continues to be a single interface with a single IP address and is independent of the instantiation of the micro-contexts (i.e., the creation of micro-contexts does not imply the creation of a dedicated external-facing L3 sub-interface for each micro-context).

The above-described mechanism may be enhanced to include Layer 4 (L4) information in the packet as part of the tag that may be used to demultiplex into a particular micro-context, resulting in a somewhat longer tag. Matching L4 information does come at the cost of a larger number of micro-contexts as each micro-context would be unique to the combination of a source IP address plus L4 (source and destination) information. By using the combination as a tag to demultiplex into the appropriate micro-context, the match on source IP address as well as traffic type (i.e., a classifier) may be achieved.

A fourth functional block encompasses the multiplexing of micro-contexts into a parent VRF. For example, all of the micro-segments that are associated with a parent VRF as micro-segments do not replace VRFs but exist within them. All of the routing information is actually contained in the VRF, and the micro-segments are used solely to determine if a packet should be forwarded or not.

Once a packet is assigned a micro-context and a destination lookup is completed in the micro-context, a packet will be handled in one of three ways. First, it may be dropped if there is no destination match. Second, it may be linked to a parent VRF for routing if there is a destination match. And third, it may be linked to a re-direct route in the parent VRF if there is a destination match and the action is re-direct.

Implementations of the above may approach the creation of the state in the micro-contexts and a parent VRF differently. For example, one approach may link partial entries in the micro-context to full route entries in the parent VRF, through which a match in the micro-context would result in a pointer to a full route in the parent VRF.

A fifth functional block encompasses route population and forwarding table state reduction. One concern with aspects of the first technique concerns the proliferation of routes that are stored multiple times across multiple micro-contexts. Two approaches may be employed to minimize this impact (all the while keeping in mind that the main goal is moving the lengthy ACE state to the forwarding table). A first approach includes the use of partial entries in the micro-context and pointers to the full state in the parent VRF (as described above in connection with the fourth functional flock). Under a second approach, the routes may be procured on-demand rather than pushed everywhere.

The on-demand approach implies that both the micro-context and the parent VRF do not have any routes cached at time zero. When a packet arrives for a new destination, a

lookup in a demand-based routing system may be initiated (e.g., a map-request to a LISP mapping system). The mapping system may respond based on the policy. In order to do so, the map-request must include the destination IP address along with the source IP address so that the policy may be evaluated in the mapping system in order to respond with a valid route or respond with a deny action. The response may be used to populate the micro-context that originated the request as well as the parent VRF. If an entry already exists in the parent VRF, the entry will be refreshed. When a deny action is met, preexisting entries in the parent VRF will not be overridden, but the route would be excluded from the requesting micro-segment in order to effectively render the policy without impacting the routing for other sources.

Finally, a sixth functional block encompasses the forwarding of traffic at egress. No special considerations are required for traffic that is being decapsulated. All of the enforcement is done at ingress and once a policy is enforced the regular forwarding ensues.

It is important to note that the first technique, as described in the above narrative, does not attempt to address the problem of enforcing micro-segmentation within or across VRFs, nor does it map subnets to SGTs or steer traffic towards security enforcement service nodes. However, such steering is possible, such as steering the request to a portal of a 'deny' splash page using a returned next hop Routing Locator (RLOC). In some instances, this RLOC could also be a security device, inserted in the path via policy. Aspects of the first technique focus on providing a mechanism for the scalable rendering of access control policies on a router or switch by leveraging the hardware forwarding tables (which are very large) and avoiding the use of the security TCAM-based tables (which are very small and limited). In contrast, some existing solutions render access control policies as Security Group Access Control Lists (SGACLs) in the (limited) security TCAM-based tables.

Additionally, under the first technique the rulesets may be large, but they are anticipated fitting into the forwarding tables. Alternatively, the rules may be stored in the control and management plane (i.e., a cloud) and computed there, with the result (e.g., a prefix, or no prefix, or redirect) being sent to the endpoint VRF table and stored in M-trie rather than in TCAM. This leads to the selective on-demand distribution of rules, which significantly reduces the amount of state at any particular enforcement point and provides

the technique with solid horizontal scale attributes. The prefixed are retrieved on an on-demand basis, further increasing the scale as compared to a SGACL push and limiting the rules that are required at an endpoint. The rules may be considered host routes and it could be argued that due to the demand-based nature, the technique can scale as user endpoints (e.g., devices) are accessed.

The implementation of the forwarding tables (specially the hardware implementation) may be optimized in the future with this concept in mind by creating an indexing mechanism that is lighter and nimbler than the one that is currently used to partition VRFs. However, this would represent a scaling optimization and it is not a hard requirement for the implementation of aspects of the first technique.

Aspects of the first technique were previously explicated through six functional blocks. In connection with those functional blocks, it is important to note the following functionalities – the assignment of a dedicated forwarding context for each source (e.g., user); the programming of constrained forwarding into the per-user context according to the access control policies; the translation of access control policies into contexts and "pseudo-routes;" the decoupling of the dedicated forwarding contexts from the overall forwarding table (this is not a VRF in the routing sense); a mechanism for the demultiplexing of plain IP traffic into the appropriate micro-context; the multiplexing of the micro-contexts (with the pseudo-routes) into full blown VRFs (with proper routes); and a demand-based model that leads to the reduction in the forwarding state that would be created at the enforcement points.

Turning to the second technique, as referenced above, aspects of this technique support, among other things, a disaggregation of the components of the policy and a distribution of the evaluation of the different components of the policy to places in the network where the amount of state that is required for the evaluation of each component of the policy is minimized due to the nature of the path that different traffic flows may follow. For example, source and destination addresses may be matched at the branches (i.e., the spokes) of a wide area network (WAN) while traffic types may be matched at the hub of the same WAN, thus reducing the number of variables that must be evaluated at the points of traffic convergence while distributing the evaluation of large state tables (comprising,

for example, source and destination IP addresses) to the branches and spokes where the state is naturally scaled horizontally.

Consequently, a canonical access policy may be disaggregated into the following components or variables – a source endpoint address, a destination endpoint address, a traffic type (which is usually the L4 classification criteria of a source port, a destination port, and a protocol but which could be expanded to include L4 through Layer 7 (L7) criteria depending upon the capabilities of the network device), and an action (such as permit, deny, redirect, etc.).

With a policy disaggregated into its components, aspects of the second technique support the evaluation of the components in phases with the phases being completed at different network locations where the amount of state that is required to evaluate the corresponding variables may be minimized. In order to complete such a phased evaluation (or enforcement) without a loss of context, aspects of the presented technique support the use of tags that identify the context that was derived at an initial phase so that it may be leveraged in a subsequent phase of the enforcement process.

While the above description focused on standard access control policies and employed two phases, it is important to note that the second technique may be expanded to other policies (such as traffic engineering policies) and may involve more than two phases. In the context of access control policies, aspects of the presented technique enable the enforcement of whitelist policies in two phases. Under a whitelist policy access is denied by default and permission must be explicitly stated as part of the policy.

As noted above, aspects of the second technique may encompass two phases of evaluation and enforcement. The first phase matches on source and destination endpoint addresses. This phase focuses on identifying the contract that governs the connection between the matched source and destination. A contract identifier (e.g., a Policy-ID) may be associated with the flow at this stage and no policy may be enforced unless the whitelist implies a deny action. Further, the second phase parses the remaining parameters in the tuple of the packet header (beyond the source and destination IP addresses) to complete the evaluation of the disaggregated policy. This phase evaluates the traffic type (e.g., L4 information) in the packet header and compares it to the L4 profile that would match the Policy-ID that was derived in the first phase.

The enforcement elements in Phases 1 and 2 must have knowledge of the Policy-IDs and how the Policy-IDs are mapped to different values of the policy components and variables that they are to evaluate. This implies that the policy is defined as a series of "contracts," each of which specify a combination of source and destination L3 and L4 information and an associated action. Each contract may be identified with a Policy-ID, which may in turn have a contract tag value. While a contract tag value may be identical to the policy-ID value, a possible difference between the Policy-ID namespace and the tag namespace is highlighted as it may allow for implementation flexibility. The policy orchestrator or controller must distribute the different components of the contracts to the different enforcement points. In doing so, the context of the Policy-ID must be associated with all policy components that are distributed. Different enforcement points will receive different portions of the disaggregated policy (overall, they are linked together by the Policy-ID as the policy is distributed). For the purposes of distributed enforcement, a Phase 1 enforcer must include the contract tag corresponding to the Policy-ID that is identified on each packet that it forwards. A Phase 2 enforcer will receive the packets and evaluate the remaining policy components in the context of the Policy-ID that is specified by the contract tag.

Aspects of the second technique may be further explicated through a simple heuristic. For example, consider that a match may be attempted first on a source, which if successful will point to a subset of the destinations that are permitted. Next, a match may be attempted on a destination (from the subset in the previous step). If no match is found, then the process may be exited and a default action (such as deny in a whitelist model) may be applied. A successful source-destination match will result in a pointer to a Policy-ID which may then be encoded in the traffic that is forwarded in a distributed system as a contract tag. The Policy-ID will point to a list of traffic types (defined by an L4-L7 tuple), which may be referred to as a contract, which specifies the different traffic types that will be matched for the particular Policy-ID. Contracts may also specify the action that is to be taken upon a match.

Next, a match may be attempted on one or more of the entries in a contract based on the Policy-ID and the policy component left to the second phase (i.e., the L4 port

numbers in the above example). Finally, a policy action (such as permit, redirect, or deny) for the matching entry may be enforced.

The matching and enforcing functions may be distributed to the routers in which the size of the memory tables with the possible matches can be minimized. Generally, the match on source-destination IP addresses would be done at an ingress router (e.g., a WAN branch router) while the match and enforcement on the entries within a contract would be done at an egress router (closer to the destination at a hub aggregation site in a WAN). Such an approach has the benefit of leveraging the horizontal scale that a network naturally achieves in a fan-out topology.

It is important to note that whenever a policy is disaggregated and distributed there are extra challenges concerning the monitoring and the correlation of the information for visibility purposes. Using the example that was presented above, distributed monitoring (according to aspects of the second technique) may work in the following manner.

When a source is matched and it points to a subset of destinations, there are two cases of interest. In a first case, permission is allowed between the source and a destination at a branch resulting in a Policy-ID. The capture of monitoring information at the branch for this traffic may be ignored since it is known when this traffic will arrive at the hub where the application of a policy based on the derived Policy-ID allows for the capture of the (source, destination, application, policy, etc.) monitoring information. As a result, this is a positive case for monitoring and visibility purposes as no information in the packet is lost.

In a second case, permission is denied between the source and destination resulting in a traffic drop at the branch. In this case the traffic is being dropped before an L4 or L7 rule as the decision to drop the packets occurs at the branch router based on partial information. As a result, this is a problematic case and there are two solutions that may be considered.

A first solution encompasses depicting the chain of visibility, where it is shown how the traffic was dropped (i.e., based on a source and a destination at a branch, and then applying a Policy-ID at a hub). Thus, a user will still obtain some visibility (though in this case the user loses some amount of granularity and visibility). A second solution, which is described below, addresses the full visibility problem.

Under a second solution, for dropped traffic at a branch, application statistics (including a source, a destination, an application type, and other required information) may be collected using a packet inspection facility and then exported to a controller. Using a correlation between the tuples that were learned from the branch and the policy rules it is possible to match the statistics and add to the data that was derived from the hub and thus obtain an aggregate view on the controller leading to fully visibility and monitoring. By employing the extra correlation logic that was described above it is possible to obtain a highly scalable policy with limited resources along with monitoring and visibility.

In summary, in support of long access control policies various solutions have been provided herein through several techniques. A first technique supports, among other things, the scaling of the access control entries (ACEs) in a network for specific deployment by converting an ACE to a route control entry. A second technique supports, among other things, the disaggregation of access control policies and the efficient distribution of their enforcement. According to this technique, access control attributes may be evaluated at different network locations to optimize scale by localizing the evaluation of matching criteria to the places in which it requires the minimum amount of state creation and maintenance. A policy may be disaggregated at the orchestrator, its components may be distributed to the different enforcement points, and a tagging mechanism may be used to unify the policy as its elements are dispersed across multiple enforcement points.