November 2020

METHOD TO SECURE TELEMETRY LOGS IN A MULTI-TENANT ENVIRONMENT

Robert Barton

Vinay Saini

Cesar Obediente

Jerome Henry

Amit Singh

METHOD TO SECURE TELEMETRY LOGS IN A MULTI-TENANT ENVIRONMENT

AUTHORS:
Robert Barton
Vinay Saini
Cesar Obediente
Jerome Henry
Amit Singh

ABSTRACT

Modern service/hosting models utilize the concept of a multi-tenant infrastructure to share common network services while providing secure logical network segmentation between tenant customers. An on-going problem with network telemetry tools in multi-tenant and shared environments is that often they do not provide data confidentiality between tenants. Presented herein are techniques in which an access policy to telemetry logs can be generated in a multi-tenant environment through the use of role-based policy tokens that are attached to log entries as they are received at a collector.

DETAILED DESCRIPTION

Data Telemetry is an essential part of modern networks. Today, users and applications are spread across multiple public and private infrastructure domains and/or clouds and can leverage different underlay and overlay networks. Modern Anything-As-A-Service (XaaS) and hosting models utilize the concept of a multi-tenant infrastructure to share common network services while providing secure logical network segmentation between tenant customers.

An on-going problem with network telemetry tools in multi-tenant and shared environments (such as Netflow and Syslog collection) is that they do not provide any form of data confidentiality between tenants. Rather, they ensure that only trusted users can access telemetry data that belongs to just the trusted users or their organization/department. This makes telemetry data sharing complex and often insecure in hosting multi-tenant environments (imagine, for example, trying to parse logs for multiple customers who share the same underlying infrastructure).

In fact, even within a single Information Technology (IT) department, not all users may necessarily have rights to view the telemetry logs for all departments and parts of the network depending on what department a user may be part of, their security clearance level, and/or compliance regulations.

Today, there are two methods to access telemetry data for customers in a multi-tenant, hosted, or shared public cloud environment:

1. Telemetry services are provisioned on a per-tenant basis, which is more secure, but increases overall cost of the service; and
2. Shared Telemetry services for all tenants, which is cost effective, but is susceptible to the security concerns mentioned above.

This proposal introduces novel method to provide Zero Trust telemetry policy, ensuring that only trusted users can view telemetry logs in either single or multi-tenant environments. This method will provide the necessary security to protect end-user data from attackers.

To realize this method, a Telemetry Policy Service (TPS) is defined in which the TPS controls the policy regarding which users have access to what types of logs, based on a variety of criteria, such as customer or tenant identity (ID), location, context, etc. The TPS is also responsible for issuing tokens to a logging server, which will be appended to each log entry to correctly identify each log entry so an access/retrieval policy may be enforced. The TPS has the main function to create and maintain a set of policies determining who can access what telemetry log entries.

Consider an operational example consisting of the following steps:

Step 1: When a new customer or virtualized tenant network is deployed, the owner/customer defines a policy identifying whom should be given access to the telemetry logs for viewing and/or retrieval. For example, log retrieval criteria may be based on one or a set of conditions, such as:

1. Customer/Tenant ID Policy: A customer/tenant ID policy can be used to allow anyone with the approved user/group ID to access all logs for a specific customer or tenant network service.

2. Location Policy: A location policy could restrict telemetry log access based on physical or cloud service location. For example, certain countries require data sovereignty. This function will enable only that certain users be allowed to view the network telemetry based on locations.
3. Network Function Policy: For a network function policy, users may be allowed to view telemetry for only certain network functions. For example, a policy could be generated in which certain users can view only overlay, underlay, firewall, and/or any other types of network function telemetry logs.
4. Cloud Service Provider policy
5. Log type: For a log type policy, users may be allowed to access telemetry matching certain flags (e.g. Facility X, video conferencing traffic, Wi-Fi associations, etc.).
6. Context: A context poly may be any type of context in addition to those mentioned above.

Step 2: For each customer/user group and policy entry, the Telemetry Policy Service (TPS) creates a policy Token. For example, telemetry logs generated by a virtual firewall service in a Seattle Data Center for a user group network named "IT Admin" could use three policy Tokens: (1) Tenant Token = IT Admin, (2) Location Token = Seattle, (3) Network Function Token = Virtual Firewall. The TPS administrator is in charge of the tokens and can renew, revoke, delegate, etc. tokens. Various example details associated with Steps 1 and 2 are illustrated below in Figure 1.

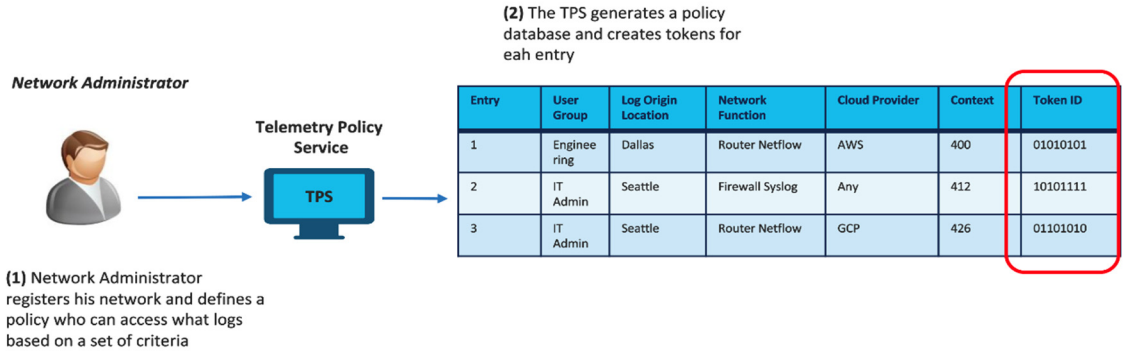

Figure 1: Example Flow Involving Steps 1 and 2

Step 3: Network devices and services begin sending telemetry log messages to a collector (e.g. sFlow, NetFlow, Syslog, firewall logs, etc. sent from a variety of network devices, including switches, hypervisors, firewalls, container service mesh nodes, etc.). The collector queries the Telemetry Policy Service (TPS) and downloads the policy Token database to determine which policy Tokens apply to each log entry.

Step 4: As log entries are received by the collector, it appends the policy Tokens to each log entry as specified by the policy. This has the effect of creating a unique descriptor for each log entry. A single telemetry log entry may have several tokens appended. In one instance, the token may act as an access authorization key. In another instance, the token may act as a decryption key. In some instances, the originating device or service that generated the telemetry logs may also query the TPS and append the correct Identity Tokens from the source. Thus, tokens are necessary for this proposal, as tokens allow for the application of granular policy to the incoming logs. Various example details associated with Steps 3 and 4 are illustrated below in Figure 2.

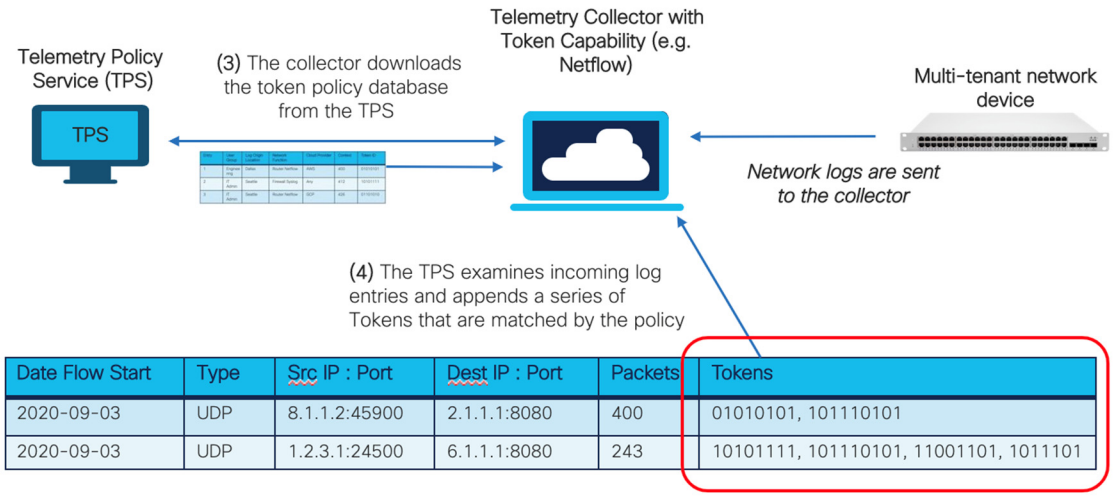

Figure 2: Example Flow Involving Steps 3 and 4

Step 5: A user logs into the collector to view telemetry logs. The collector must now authorize the user on the TPS.

Step 6: The collector queries the TPS to determine the policy Tokens for which this user has privileges. The TPS returns the set of policy Tokens.

Step 7: The Collector now parses the log entries and looks for logs entries with the Tokens for which this user has access. Only logs being requested in which the Tokens match will be retrievable. Certain users may only be allowed to view specific logs and may be restricted from others, depending on the policy. If a log entry has multiple policy Tokens, as shown in Figure 2, above, the user may have to match all of the policy Tokens. Various example details associated with Steps 5, 6, and 7 are illustrated below in Figure 3.

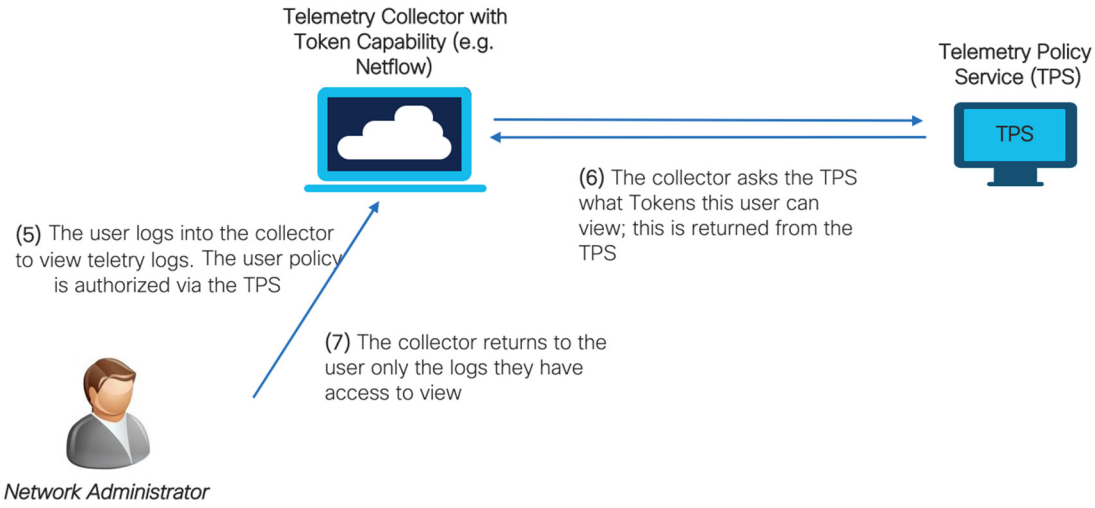

Figure 3: Example Flow Involving Steps 5, 6, and 7

With regard to protecting the actual tokens, the policy engine is authoritative and is the only one that can issue policy tokens to either a user or a collection device. Both users and collection devices must be trusted and authenticated (e.g., using a public key infrastructure (PKI)) in order to receive or use the policy tokens. Once a token is issued, it can be protected through various means, such as encryption on a local database, etc. In addition, the tokens themselves can be randomly generated by the policy engine to limit being easily predicted. Additionally, in some instances, tokens can be periodically changed

in real-time by the policy engine to maintain security (e.g., through a communication between the policy engine and the collector's local policy token database).

In summary, techniques herein provide for the ability to generate an access policy for telemetry logs in a multi-tenant environment through the use of role-based policy tokens that are attached to log entries as they are received at a collector.