Technical Disclosure Commons

Defensive Publications Series

September 25, 2018

INTERNET OF THINGS SECURITY DOMAIN NAME SYSTEM POLICY AND ANALYTICS

Grant Regan

Abdel Abdel-Halim

INTERNET OF THINGS SECURITY DOMAIN NAME SYSTEM POLICY AND ANALYTICS

AUTHORS:
Grant Regan
Abdel Abdel-Halim

ABSTRACT

Techniques are described herein to allow each Domain Name System (DNS) packet to be identified by enterprise and device at the DNS server. The DNS server may apply security rules per device and generate reports and charts on a per device level of granularity for analytics about the traffic behavior/trend for each enterprise.

DETAILED DESCRIPTION

All Domain Name System (DNS) resolution traffic, for all of a network's devices, is received by a remote DNS server all on the same Internet Protocol (IP) address. The DNS server only knows what network, but not which enterprise customer or which device on that network, is making the request based on its network IP address. Therefore, it can only apply blanket policies to all of the network devices whose traffic is received from that network address.

DNS filtering can be performed on all devices in a network and those rules are applied to all devices in that network. Internet of Things (IoT) and cellular networks are networks that are shared among multiple operators, and multiple enterprises. DNS filtering can be applied at an enterprise level but not on a device level for cellular devices or IoT devices within the operator or Mobile Virtual Network Operator (MVNO) network.

Accordingly, the techniques described herein enable an IoT network operator to apply policies and provide the capability for analytics and machine learning on a user/device level of granularity on a remote DNS server. A device level DNS filtering solution is provided with machine learning and device level analytics for IoT. An IoT network is comprised of shared nodes at the cellular operator or an operator partner's network. When using these shared nodes, the devices are similarly viewed when going to the Internet as coming from a single IP address of the network on an external firewall. In some cases, a network operator may purchase multiple instances of the firewall, group

traffic on those separate IP addresses, and apply policies on the DNS server per network IP address (enterprise customer) but the policies are still applied as a whole to all devices received from that specific network IP address. Device identifiers are included to apply security policies on individual devices while providing a mechanism for machine learning and data analytics to generate insights into individual device behaviors.

Existing next generation firewalls may be placed in the network. However, a separate firewall per IoT enterprise is required to apply the user defined policies and the dynamically updated policies defined on the remote DNS server. Yet this does not provide these enterprise customers the level of visibility at the device level provided by the solution. Additionally, in IoT enterprises that are relatively small or generate small revenue, a dedicated node in the network is not practical. The techniques described herein may be beneficial to the network operators by providing device level security policies, device behavior reporting, and allowing these IoT enterprises to scale without incurring costs by removing the requirement of having the cost prohibitive burden for upgrading to intelligent firewalls, and also operating and maintaining the additional network nodes.

These techniques require little to no network infrastructure upgrades to apply policies and provide analytics on a device level granularity using a remote DNS server. Within the operator network, the user/device is identified, the DNS packet is broken down, and the DNS packet is reassembled with the user/device information to be filtered if necessary by the remote DNS server. This may also be logged for analytics and machine learning.

Additional identifiers may be extracted from the Remote Authentication Dial-In User Service (RADIUS) call flow and included in the Extension mechanisms for DNS (EDNS) request. Those fields may be extended along with the original request to the DNS server to select the security policy on the individual device level. Analytical charts and logs may be generated for each device and enterprise.

An example method for IoT device DNS policy and analytics is provided as follows. The first step is to collect the device network and authentication records from the Authentication, Authorization, and Accounting (AAA) server and store that information in the Lightweight Directory Access Protocol (LDAP) server. The second step is to capture the DNS packet in the load balancer. The third step is to initiate a lookup mechanism for

the device identifying information. The fourth step is to disassemble the DNS packet and reassemble the DNS packet with the device identifying information. The fifth step is to send the EDNS query sent to the DNS Server. The sixth step is to respond, at the DNS server, to the DNS request while applying the security policy based on the device identifying information and also store the records for that device for device level analytics.

Figure 1 below illustrates an example call flow.

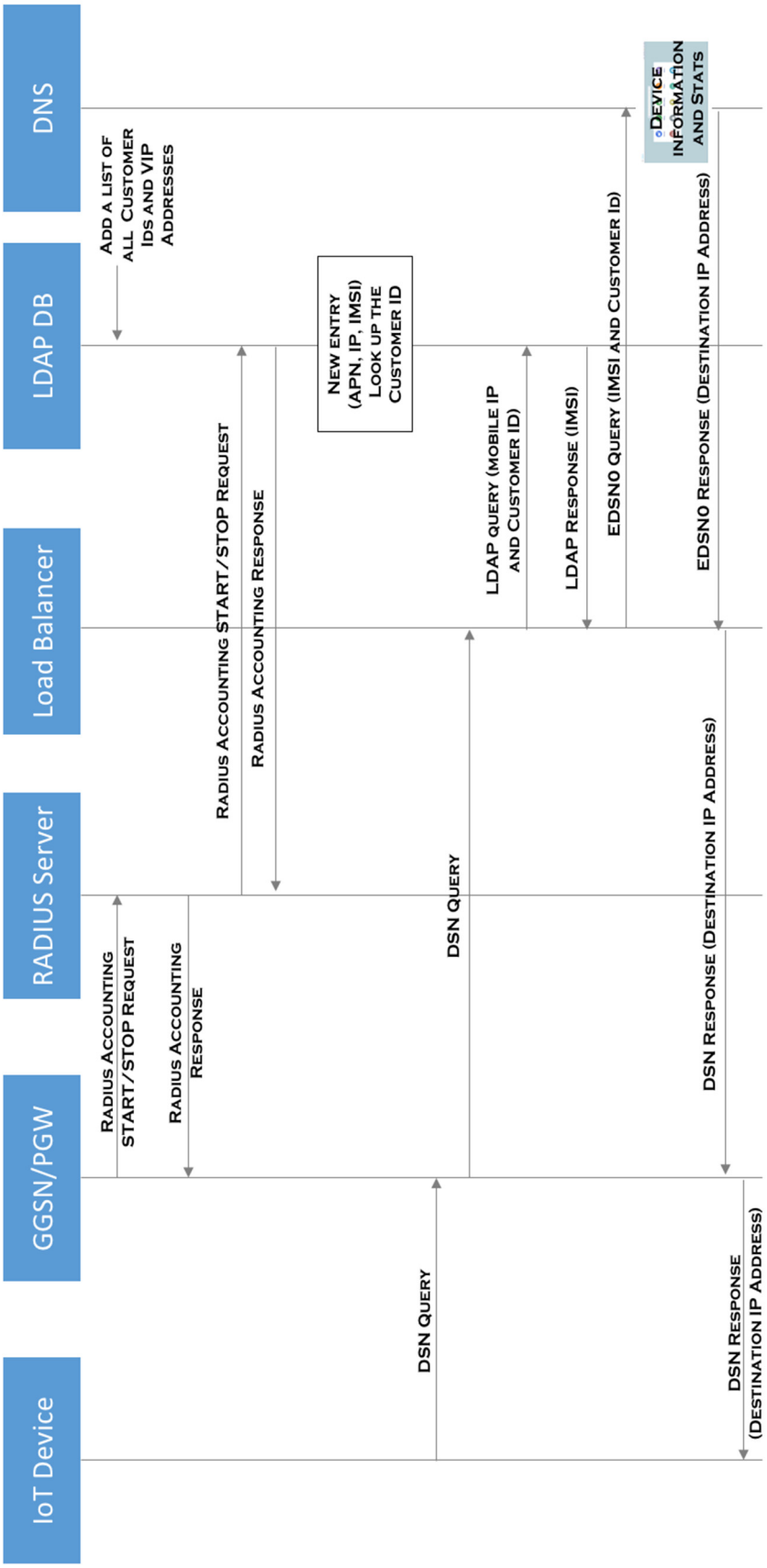

Figure 1

Figure 2 below illustrates an example network diagram.

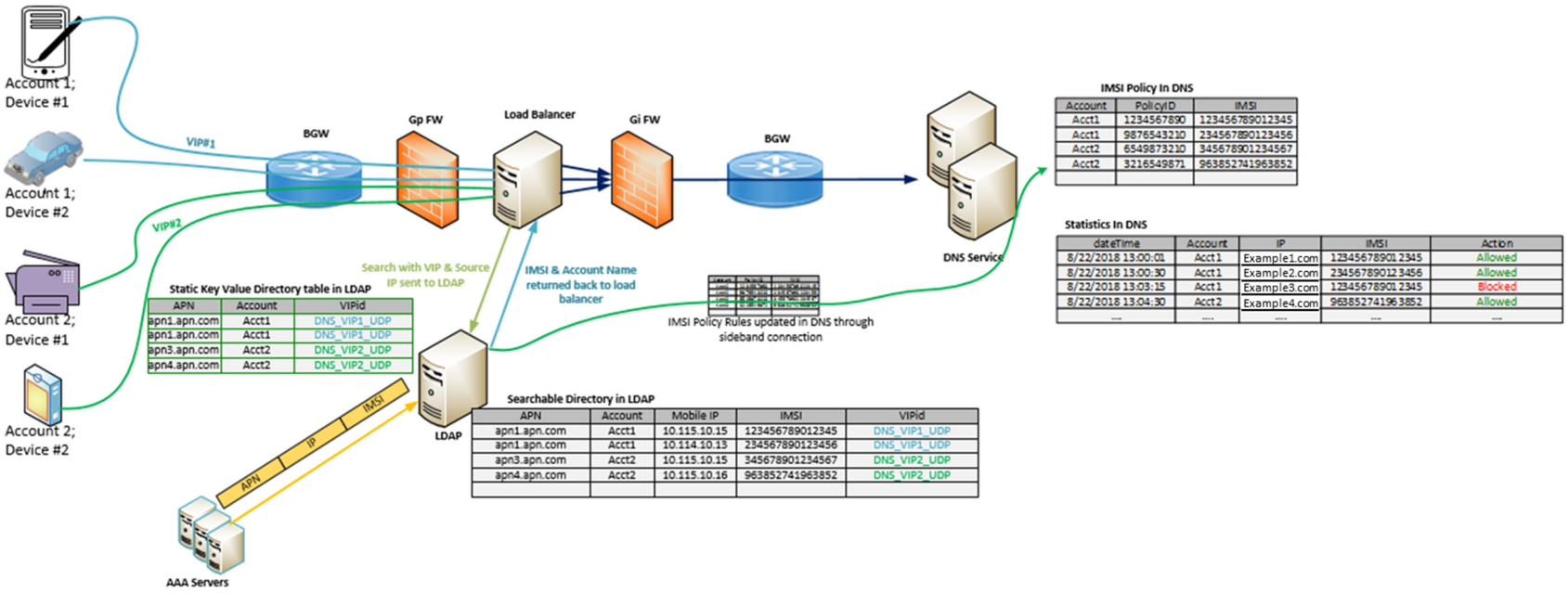

Figure 2

Figure 3 below illustrates example load balancer logs.

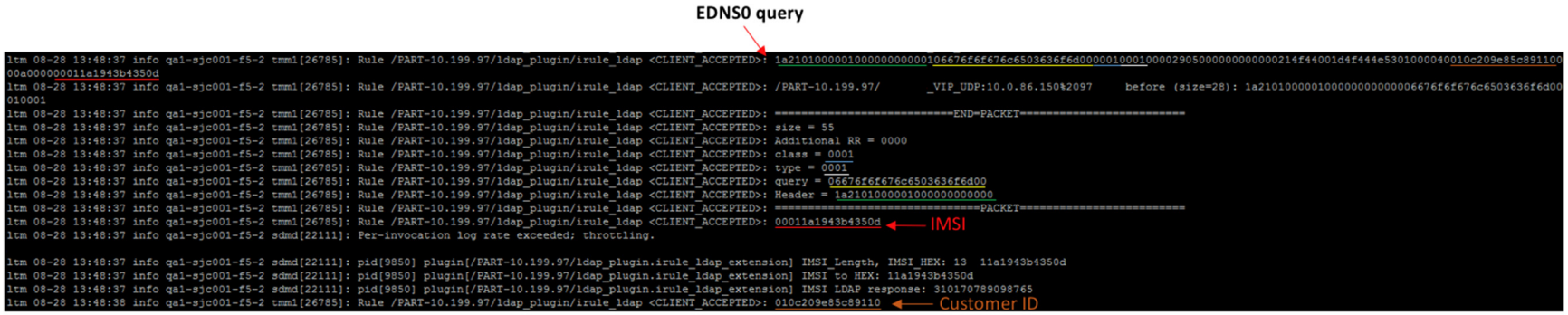

Figure 3

Figure 4 below illustrates an example tcpdump.

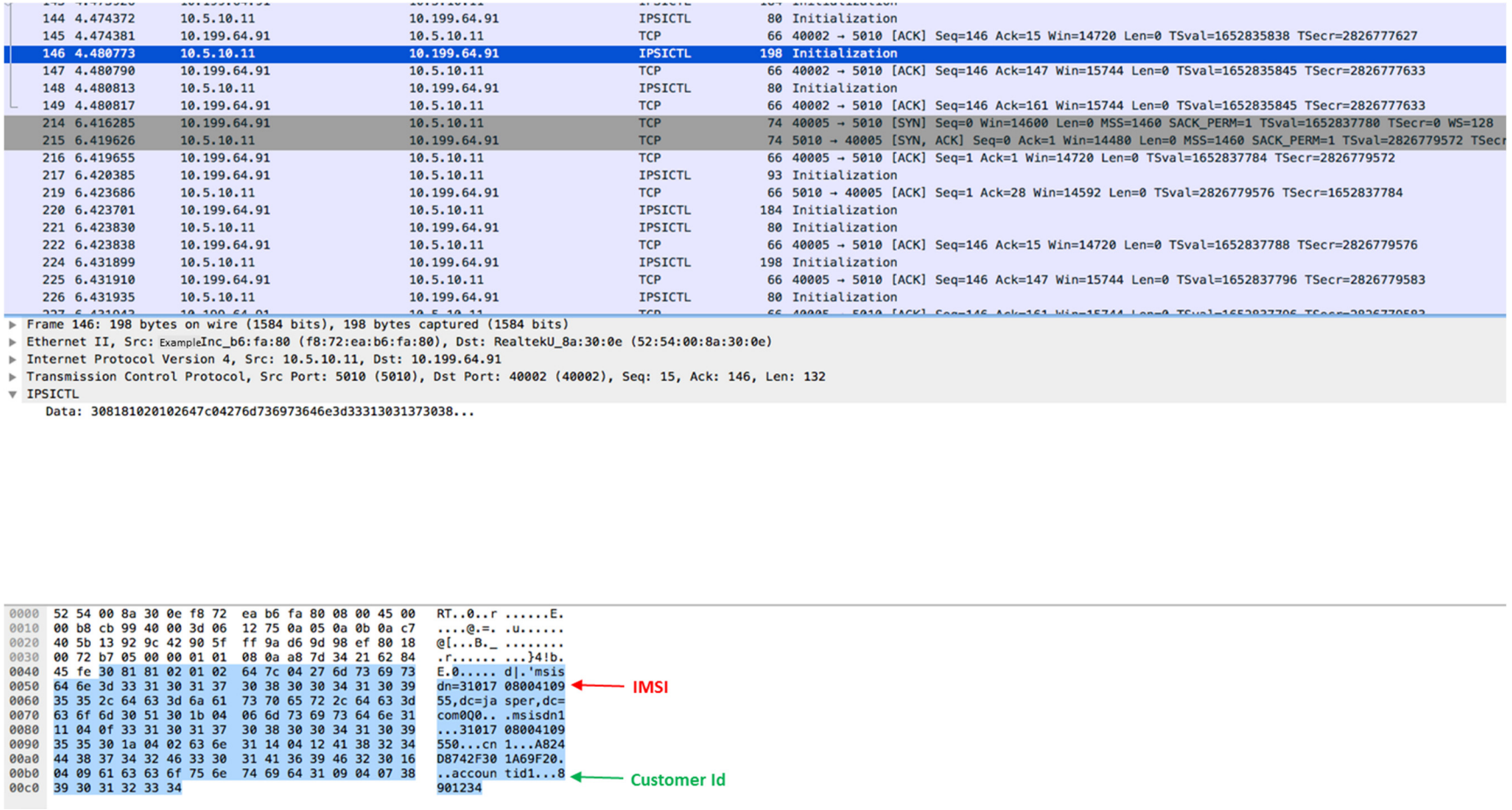

Figure 4

An identifier is added in the EDNS0 request to identify a device among many different customers without changing the DNS name or the DNS query. Also, the ability to configure security policies is provided at the subscriber/device level, and all activities are logged for analytics purposes. Current techniques change or automate the DNS name which is the identifier of the device in order to be able to access each individual device. By contrast, the techniques described herein add the information in the packet to be interpreted by the remote DNS server for analytics and policy enforcement on the individual device level.

The DNS standard (see Internet Engineering Task Force (IETF) Request for Comments (RFC) 2671) is used to attach a resource record to the DNS packet with the additional identifier without changing the DNS name or query name. This enables compatibility with any standard DNS server in the public cloud. Current techniques propose changing the DNS name within a private network to interpret different queries, whereas the techniques described herein append one or multiple identifiers to the additional section in the DNS query using the standard RFC 2671. If the query is received by a DNS server that does not comply with this standard, then it will not break the DNS flow but it may not be able to provide the additional layer of security and policy as needed. The aforementioned concept may be used along with device metadata from network nodes (e.g., AAA RADIUS servers) to extract dynamic parameters to couple them with different identifiers.

The techniques described herein are not vendor specific and may be used with any DNS service capable of supporting EDNS0. Moreover, the user is not required to use a specific firewall, and the existing network infrastructure may remain as-is with minimal network infrastructure upgrades required. Without these techniques, the user would have needed to deploy dedicated firewalls per IoT account to enable similar features. In addition, due to the amount of revenue that individual IoT accounts generate, these features cannot sustain positive revenue while having dedicated network nodes per account.

In summary, techniques are described herein to allow each DNS packet to be identified by enterprise and device at the DNS server. The DNS server may apply security rules per device and generate reports and charts on a per device level of granularity for analytics about the traffic behavior/trend for each enterprise.