Technical Disclosure Commons

Defensive Publications Series

December 04, 2018

WIRELESS NETWORK LOAD CORROBORATION USING MACHINE LEARNING (ML) BASED VIDEO ANALYTICS

Smruti Lele

Samer Salam

Ajay Madhavan

Jayesh Wadikar

WIRELESS NETWORK LOAD CORROBORATION USING MACHINE LEARNING (ML) BASED VIDEO ANALYTICS

AUTHORS:

Smruti Lele
Samer Salam
Ajay Madhavan
Jayesh Wadikar

ABSTRACT

Presented herein are techniques for correlating the output of a crowd counting machine learning (ML) algorithm, which operates on surveillance video, with observed network load to determine if a load spike is due to a valid network usage or an attacker trying to sabotage the network. The techniques presented herein include vision field classification based on access point (AP) coverage, linking of vision fields to AP coverage in DNAC UI, and consensus-based threat assessment and alerts.

DETAILED DESCRIPTION

Wireless networks deployed in public venues (e.g., airports, stadiums, *etc*.) exhibit large variance in traffic volume characteristics, depending on the number of occupants and their mobility patterns. For instance, in an airport, it is expected to see network capacity usage spiking at the access points near the gate/terminal at which an aircraft arrives. A problem with conventional techniques is that there is no way to determine whether the capacity increase is due to real users or a fake capacity load created by an attacker/jammer.

Section 1: Solution Summary

The techniques presented herein propose correlating the output of a crowd counting machine learning (ML) algorithm, which operates on surveillance video, with observed network load to determine if the load spike is due to a valid network usage or an attacker trying to sabotage the network. In general, the crowd counting algorithm will leverage out of band information from co-located camera systems or other third party security systems that provide a programmable application program interfaces (APIs), such as programmable

closed-circuit television (CCTV) control APIs, programmable camera APIs, programmable security system APIs. The techniques presented herein include vision field classification based on access point (AP) coverage, linking of vision fields to AP coverage in DNAC UI, and consensus-based threat assessment and alerts.

Wireless attackers are difficult to classify since there is no physical connection which identifies devices in the wired world. This makes the detection of such attackers all the more difficult. Moreover, unlike wired networks where attacks can be launched from a different continent, by the nature of wireless channels themselves, a wireless attacker needs to be in proximity to the access point to launch an attack. As such, the techniques presented herein address an important problem where, by correlating a wireless user location with certain vision fields, the chances of catching an attacker (if any) on camera are significantly increased. Additionally, if the attack is conducted using an RF-jammer, in a regular case there are no simple solutions except just changing wireless channels. However, using the techniques presented herein, a deterrent is created by capturing the attacker on video or pictures. This also helps to confirm the identity and location of the jammer (if any).

Section 2: Design Overview – Camera-Augmented Security System

An example deployment for a Camera-Augment Security System, as proposed herein, is shown below in FIG. 1.

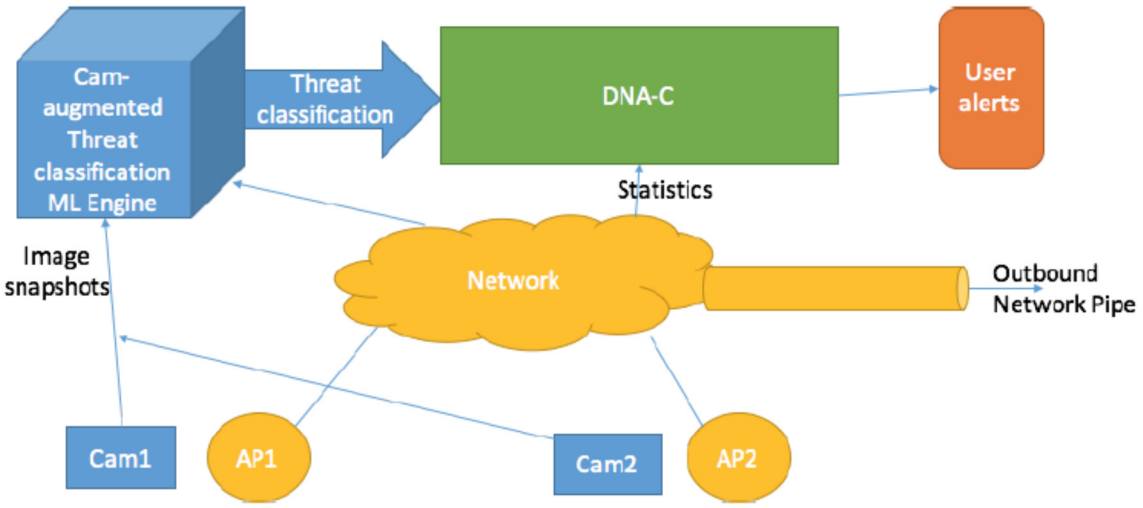

FIG. 1

As shown, the deployment includes cameras that are co-located with access points. In certain deployments, the cameras have video coverage over the typical service area of access points. The inputs from the cameras are fed into a camera-augmented threat classification engine. This ML classification engine takes into account both the images from the cameras as well as network statistics provided from the access points (e.g., via the controller). The output from the ML classification engine is fed to a controller for alerts and display. Each of the components within the deployment are explained in further detail below.

Section 3: Vision Field Correlation with AP coverage

A vision field is defined as the area on a floor which is visually seen by a camera. The first design challenge for the proposed approach is the matching of the images from the cameras (vision fields) with the coverage area of the access point. In one embodiment, the programmable cameras are coupled with access points at the time of deployment through a provisioning step. In another embodiment, an automatic discovery mechanism may be employed.

Section 4: Consensus Based Threat Assessment and Alerts

For the purpose of threat classification, the techniques presented herein use an ML approach that jointly looks at the network statistics from the access point and the image output from the corresponding vision fields of the cameras. In one example, a smart threat detection and classification architecture is designed as shown below in FIG. 2.

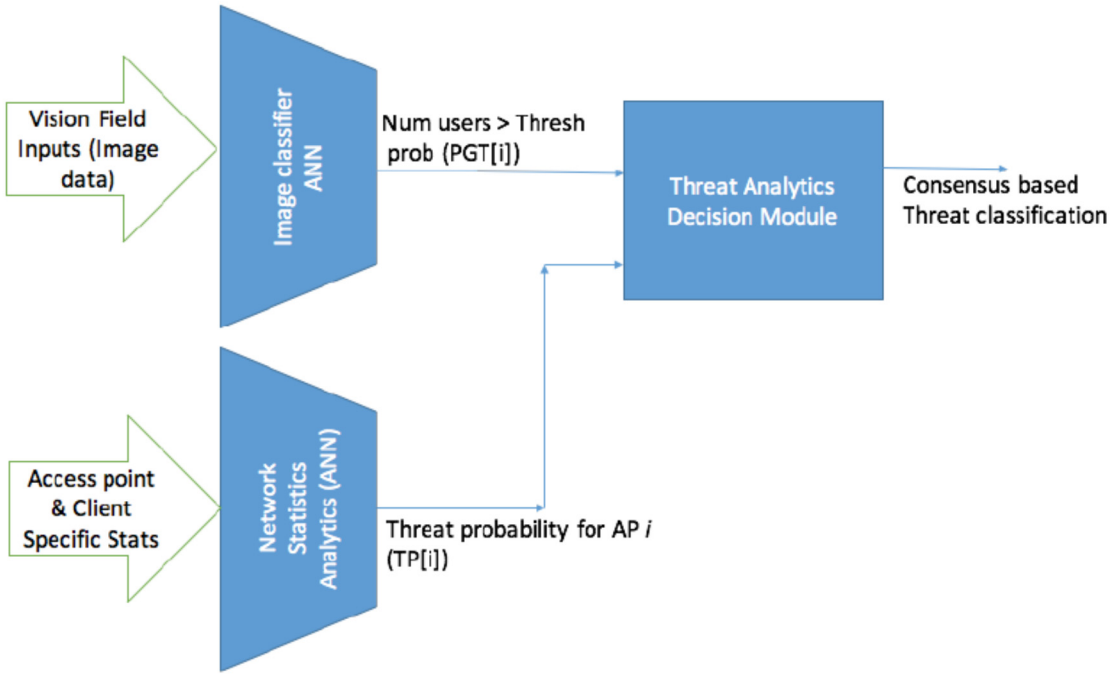

FIG. 2

In this embodiment, two (2) ML components, one for image classification and one for threat classification, are employed and run based on network statistics. The output of both of these components are probabilities of certain events which are fed into the threat analytics decision box. A description of how each of these components are setup and used is provided below.

Image Classified Artificial Neural Network (ANN)

There are a number of third party and open source neural network engines which can do image classification. In accordance with the techniques presented herein, the classifier is trained to flag images from the cameras that correlate with high usage vs those that do not. It is important to note that such an approach will also take into account usage patterns, such as differentiating a couple of users generating a huge load, a lot of users, or a jammer.

The output of this classifier is the probability outcome: Num users > Thresh prob (PGT[i])

Network Statistics Analytics ANN

Proposed herein is the use of another ANN that independently looks at network statistics, such as the number of unique MAC addresses seen in a time window, the amount of traffic passed through the access point, the ratio of uplink traffic to downlink traffic, data rates used by the traffic, *etc*. This ANN is a ML engine, which looks at these statistics and predicts if these are potentially from an attacker or not.

The neural network will use sigmoid activation functions with back propagation. These will be trained on well-known attack and benign traffic patterns seen in the field for classification.

The output of this ML engine is a probability TP[i], which indicates if the current instance of network statistics for AP[i] indicates an attack.

Threat Analytics Decision Box

The threat analytics decision module accepts the following inputs from the two previous modules for every instance of images relevant to AP[i] and corresponding AP[i] statistics:

(1) Probability outcome: Num users > Thresh prob = PGT[i];

(2) Threat probability seen at AP[i] which is denoted as TP[i]; and

(3) Traffic load at AP[i] denoted as L[i].

Outcomes (classifications and corrective actions)

- Action 1-- If it is a legitimate group of users, then no actions are taken.
- Action 2-- If it is disproportionate usage by a few users, the information could be used to throttle user limits.
- Action 3-- If an attacker is observed, this can be flagged to a system administrator and he/she may use the camera feeds to flag that as actual events or false detects.

In one embodiment, the module performs the following:

```
103
104 In one embodiment, this module does the following:
105 < For every set of images and network statistics for AP[i] >
106 if (TP[i] < UB_THRESH_TP[i]) {
107     // Most likely not an attack.
108     if (L[i] <= THRESH_L[i]) {
109         do Action1
110     } else {
111         // Operating under high load.
112         if (PGT[i] < THRESH_PGT[i]) {
113             // Few users hogging the network
114             do Action2
115         } else {
116             // Legit case: More users more load
117             do Action1
118         }
119     }
120 } else {
121     // Potential threat at AP[i], corroborate
122     if (L[i] > THRESH_L[i])
123     {
124         // Operating under high load.
125         if (PGT[i] < THRESH_PGT[i]) {
126             // Few users, high load and threat - this is an atta
127             do Action3
128         } else {
129
130         // Load is high but more users. Joint attack?
131         // Check against a more conservative threshold.
132         if (TP[i] > LB_THRESH2_TP[i])
133             do Action3
134         }
135
136     } else {
137
138         // Load is low, even though TP[i] is high.
139         // False detection
140         do Action1
141     }
142 }
```

In this case, the classifier's output can be used to compare what is observed through network statistics (e.g., approximate clients or traffic generating unique mac addresses). In this case, three simple classification buckets, such as (1) <10 users, (2) 10 to 20 users, and (3) >20 users could be generated. This output of the classifier could be used to match with what is observed in the network monitoring.

Section 5: Linking of Vision Fields to AP Coverage in DNAC UI

Once the images are matched (cameras and their outputs) with the coverage area of wireless access points, the techniques presented may employ section in the UI where a user can click an access point on the topology and see the images of the coverage area. Such an approach in the UI uniquely ties the network maintenance with smart camera outputs.

Section 6: Discussions

In general, the techniques presented herein are ML independent and are not tied to one type of ML for crowd counting and classification. In fact, in another embodiment, the techniques presented herein can separately leverage more than one third party or open source applications for this process.

Additionally, the techniques presented herein are camera hardware independent and can leverage any programmable hardware for the deployment.