Behavior-Profile Clustering for False Alert Reduction in Anomaly Detection Sensors

Vanessa Frias-Martinez Salvatore J. Stolfo
Angelos D. Keromytis
Computer Science Department, Columbia University
500 West 120th Street, New York, NY 10027
vf2001,sal,angelos@cs.columbia.edu

Abstract

Anomaly Detection (AD) sensors compute behavior profiles to recognize malicious or anomalous activities. The behavior of a host is checked continuously by the AD sensor and an alert is raised when the behavior deviates from its behavior profile. Unfortunately, the majority of AD sensors suffer from high volumes of false alerts either maliciously crafted by the host or originating from insufficient training of the sensor. We present a cluster-based *AD sensor that relies on clusters of behavior profiles to identify anomalous behavior. The behavior of a host raises an alert only when a group of host profiles with similar behavior (*cluster of behavior profiles*) detect the anomaly, rather than just relying on the host's own behavior profile to raise the alert (*single-profile *AD sensor). A cluster-based AD sensor significantly decreases the volume of false alerts by providing a more robust model of normal behavior based on clusters of behavior profiles. Additionally, we introduce an architecture designed for the deployment of cluster-based AD sensors. The behavior profile of each network host is computed by its closest switch that is also responsible for performing the anomaly detection for each of the hosts in its subnet. By placing the AD sensors at the switch, we eliminate the possibility of hosts crafting malicious alerts. Our experimental results based on wireless behavior profiles from users in the CRAWDAD dataset show that the volume of false alerts generated by cluster-based AD sensors is reduced by at least 50% compared to single-profile AD sensors.*

1 Introduction

Anomaly Detection (AD) sensors are technologies designed to detect anomalies in the behavior of a host. AD sensors compute the behavior profile of a host by using a set of statistical features that characterize its typical use of services. Any behavior that deviates from the behavior profile is deemed anomalous and an alert is generated [3]. While AD sensors provide an effective way of detecting anomalous behavior, most are plagued with high volumes of false alerts. These false alerts may originate either from the AD sensor itself because of poor or insufficient training, or else from a host that is maliciously trying to generate alerts.

Redundancy has been proposed as a means to reduce the volume of false alerts [1], [13], [16]. Each network host is assigned multiple sensors that observe and model its behavior through different correlated parameters. Alerts are generated whenever an agreement is reached among multiple sensors. By redundantly modeling the same behavior with different types of sensors, the generation of false alerts either maliciously crafted or due to insufficient training is greatly reduced. Nevertheless, redundancy tends to overload the host since the computation of profiles takes away computing cycles from real host applications.

Collaboration through alert sharing has been explored as a way to correlate alerts to detect coordinated attacks, indirectly reducing the number of false alerts [10], [11], [14]. AD sensors across one or multiple domains share alerts amongst themselves and then perform a correlation analysis to understand their nature. Contemporary alerts may be related to the same event, possibly an attack. Alerts not found by any other collaborating host should probably be considered false alerts. This approach can significantly reduce the volume of false alerts when coordinated attacks take place; however, it leaves each sensor on its own during isolated attacks on single hosts since other sensors in the network do not observe the attack. We extend the idea of alert sharing to behavior-profile sharing as a way to reduce the number of false alerts during isolated attacks.

In this paper, we present a *cluster-based* AD sensor that relies on the use of *clusters of behavior profiles* to perform the anomaly detection. As introduced in [4] and [5], a cluster of behavior profiles constitutes a collection of profiles

from hosts behaving similarly. In a cluster-based AD sensor, the behavior of each host is compared against its own profile as well as against the profiles of the hosts within its own cluster of behavior profiles. An alert is generated only when all the profiles in the cluster of behavior profiles agree on the anomalous nature of a certain host. In contrast, a *single-profile* AD sensor compares the behavior of the host only against its own behavior profile. Consequently, a cluster-based AD sensor provides a more informed decision for the generation of alerts than a single-profile AD sensor since a group of profiles representing a similar behavior offers a more robust description of behavior normalcy.

Apart from the design of the sensor, we envision its actual deployment in a network as a key component of our proposal. In particular, we focus on the deployment in small- to medium-sized networks e.g., enterprise or university. The deployment of AD sensors in a network can follow a *fully distributed* or a *hierarchical* architecture. Fully distributed architectures such as CSM [16] locate at least one AD sensor on each host. Each sensor is then responsible for the behavior-profile modeling, the alert generation, and the alert analysis. On the other hand, hierarchical approaches separate the profile computation and alert generation (performed at the host) from the alert analysis that takes place in higher-ranked monitors or correlating hosts [1], [12]. Unfortunately, the placement of the alert generation on the host opens the possibility that a compromised host may maliciously craft an alert. We propose an architecture where each network switch is responsible for computing and updating the behavior profiles of all the network hosts in its subnet. These behavior profiles are then exchanged among switches and clustered to identify the clusters of behavior profiles in the network. Armed with these clusters, each switch can perform a cluster-based anomaly detection for each of the hosts in its subnet. More importantly, by performing the anomaly detection at the switches, hosts are prevented from maliciously crafting false alerts.

Throughout the paper, we make four key assumptions. First, all communications are secure. As a result, behavior profiles cannot be modified during exchanges. Second, hosts openly share their profiles with switches but not with other hosts in the network. Third, the AD sensors compute behavior profiles of hosts based on their network activity. Finally, we assume that switches are more robust to attacks than individual hosts. Our main contributions are:

- A cluster-based AD sensor that uses clusters of behavior profiles to achieve a more informed decision for the generation of alerts. In contrast to single-profile AD sensors, a group of profiles representing a similar behavior constitute a more robust description of behavior normalcy for a host.
- An architecture where the behavior-profile modeling of each host and its anomaly detection is performed at its closest switch. As a consequence, the possibility of false alert generation from compromised hosts is eliminated.
- An evaluation with 100 real user behavior profiles computed from traces of wireless traffic captured at *Dartmouth University* (CRAWDAD repository).

The paper is organized as follows: Section 2 describes the cluster-based AD sensor, Section 3 presents an architecture designed for the deployment of cluster-based AD sensors in a network, Section 4 describes the experimental results and presents a comparative analysis between single-profile and cluster-based AD sensors, Section 5 provides estimates of the network bandwidth requirements associated with the deployment of cluster-based AD sensors, and Section 6 summarizes related work. Lastly, conclusions and future work are covered in Section 7.

2 Cluster-Based Anomaly Detection Sensor

Current network-based AD sensors compute the behavior profile of a host based on its normal network activity. In order to detect anomalous activities, the AD sensor compares the input or output traffic to or from a host against its behavior profile. Any behavior that deviates from the profile is considered anomalous and generates an alert. We shall refer to this type of AD sensor as single-profile since it only compares the traffic against its own behavior profile. While this approach is effective in detecting anomalous behavior, it suffers from high volumes of false alerts mainly generated from insufficient or poor training data.

In order to alleviate this shortcoming, we introduce a cluster-based AD sensor that relies not only on the host's own profile but also on the behavior profiles of other hosts that share a similar behavior in the network. The advantage of this approach is that it provides a broader definition of normal behavior that compensates for insufficient or poor training data in single-profile AD sensors. As a result, cluster-based AD sensors can potentially lower the volume of false alerts. Similar to a single-profile AD sensor, a cluster-based AD sensor computes the behavior profile of a host based on its network activity. But rather than relying only on its own behavior profile for the anomaly detection, it makes use of clusters of behavior profiles. A cluster of behavior profiles represents a set of host profiles that share similar network behavior [5]. For instance, one could differentiate between a cluster of behavior profiles consisting of highly active hosts with a large number of frequent connections to different IPs and another cluster made up of less active hosts that connect to fewer IPs with lower frequency. A cluster-based AD sensor performs the anomaly detection for a host by comparing the traffic exchanged by the host

(a) Single-profile AD sensor for traffic to or from host *X*.

(b) Cluster-based AD sensor for traffic to or from host *X*.

Figure 1. Single-profile versus Cluster-based AD sensor.

against its own behavior profile and against the behavior profiles in the cluster where the host is a member e.g., the cluster-based AD sensor of a highly active host would compare the traffic exchanged by the host only against the profiles of other highly active hosts. An alert is generated only when all the profiles in the cluster of behavior profiles agree on the anomalous nature of the traffic as opposed to single-profile anomaly detection that generates an alert when the traffic is considered anomalous only by its own behavior profile.

In practical terms, whenever a host i sends or receives traffic, its sensor S_i verifies whether it is within one standard deviation σ of its behavior profile p_i and all the other profiles p_m of the cluster c where p_i is a member (see Equation 1). The sensor compares the traffic t against each profile p_m in the cluster and emits a partial decision $D_{p_m}(t)$ regarding the anomalous or normal nature of the traffic. If all the partial decisions agree on the anomalous nature of the traffic, the sensor generates an alert $S_i = 1$. Otherwise, the sensor deems the traffic normal $S_i = 0$. Hence, S_i is defined as follows,

$$S_i = \begin{cases} 1 & \text{if } \forall p_m \in c \;\; D_{p_m}(t) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$D_{p_m}(t) = \begin{cases} 0 & \text{if } p_m - \sigma < t < p_m + \sigma \\ 1 & \text{otherwise} \end{cases}$$

Figures 1(a) and 1(b) highlight the differences between a single-profile and a cluster-based AD sensor for a network host *X* that shares similar behavior with network hosts *Y* and *Z*. In a single-profile AD sensor, the traffic to or from host *X* is only checked for anomalies against its own profile p_X. In this instance, anomalous traffic is detected and an alert is generated by the sensor (see Figure 1(a)). The cluster-based AD sensor, on the other hand, compares the traffic to or from host *X* against the behavior profiles p_X, p_Y, and p_Z. While the sensor generates an alert comparing the traffic to behavior profile p_X, the traffic is not deemed anomalous when compared to behavior profiles p_Y and p_Z (see Figure 1(b)). As a result, the alert from host *X* is dismissed as a false alert and the traffic is deemed normal. Such an alert could have been due to insufficient or poor training of p_X and the cluster-based AD sensor eliminates it successfully. In other words, a cluster-based AD sensor can also be understood as an automatic online tuning for single-profile AD sensors whereby the profiles within a cluster of behavior profiles determine the guidelines for a more informed anomaly detection.

3 Architecture

After describing the principles of cluster-based AD sensors, the next crucial step is the design of an architecture that maximizes their efficiency on an actual network. In our architecture, each network switch has one cluster-based AD sensor for each of the hosts within its subnet. Initially, the switch computes the behavior profiles of each of its hosts using their individual cluster-based AD sensors. Alternatively, the computation of the profiles could be performed on the hosts themselves and then submitted to its closest switch to avoid overloading the switch with excessive computation. Upon computation of the behavior profiles from all the hosts in its subnet, each network switch submits the profiles to a central router. This central router then executes a clustering algorithm to identify clusters of behavior profiles. Each resulting cluster is composed of profiles from

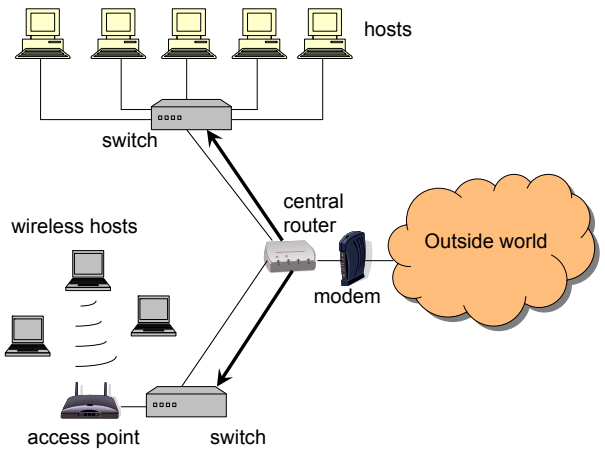


Figure 2. Schematics of the architecture. The arrows depict the broadcast of the clusters of behavior profiles from the central router to all switches in the network.

network hosts that share similar behavior.

These clusters of behavior profiles are then broadcasted to each switch such that a definition of normal types of behavior in the network is known all across. Figure 2 shows the schematics of the architecture. In particular, the central router broadcasts the computed clusters of behavior profiles to all the switches in the network. As shown, the architecture that we present applies to both wired and wireless networks. In the wired case, hosts are directly connected to the switch. On the other hand, wireless hosts communicate with the switch through an access point. Throughout, we assume that the switch has the capability to perform the computation of profiles as well as the anomaly detection, or else that there exists a group of dedicated hosts connected to the switch performing these activities. It is important to note that by performing the anomaly detection at the switches, hosts are prevented from maliciously crafting false alerts.

Once the clusters of behavior profiles have been broadcasted by the central router, each network switch proceeds to update the cluster information. Each cluster-based AD sensor S_i on the switch is updated with its respective cluster of behavior profiles i.e., the behavior profiles of the hosts that share a similar behavior with host i. At this stage, the cluster-based AD sensors are ready to perform the anomaly detection on the traffic exchanged among network hosts. The traffic to or from each host is compared against its own profile as well as against the profiles of the hosts within its own cluster of behavior profiles. An alert is generated only when all the profiles in a cluster of behavior profiles agree on the anomalous nature of the traffic.

Figure 3(a) depicts a traffic exchange between two hosts (from host *A* to host *E* in this case). Each switch has a cluster-based AD sensor for each host i in its subnet. These cluster-based AD sensors S_i store the respective cluster of behavior profiles that contains the profiles of the hosts with similar behavior to i. In this instance, four different clusters of behavior have been identified by the central router (see Figure 3(b)) and broadcasted to each switch. The anomaly detection that takes place when host *A* sends traffic to host *E* proceeds as follows: traffic from host *A* goes through *switch1* which forwards the traffic to the cluster-based AD sensor of host *A* (S_A). This sensor checks the normalcy of the traffic by comparing it against its own behavior profile p_A and against all remaining profiles of the cluster where host *A* is a member i.e., *cluster2* that contains the profiles p_B and p_C. From the comparison with the profiles, the sensor emits three partial decisions D_{p_A}, D_{p_B}, D_{p_C} regarding the nature of the output traffic from host *A* (see Equation 2). If all the partial decisions agree on the anomalous nature of the traffic, the sensor generates an alert $S_A = 1$. Otherwise, the sensor deems the traffic normal $S_A = 0$.

$$S_A = \begin{cases} 1 & \text{if } \forall p_i \in cluster2 \;\; D_{p_i}(t) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

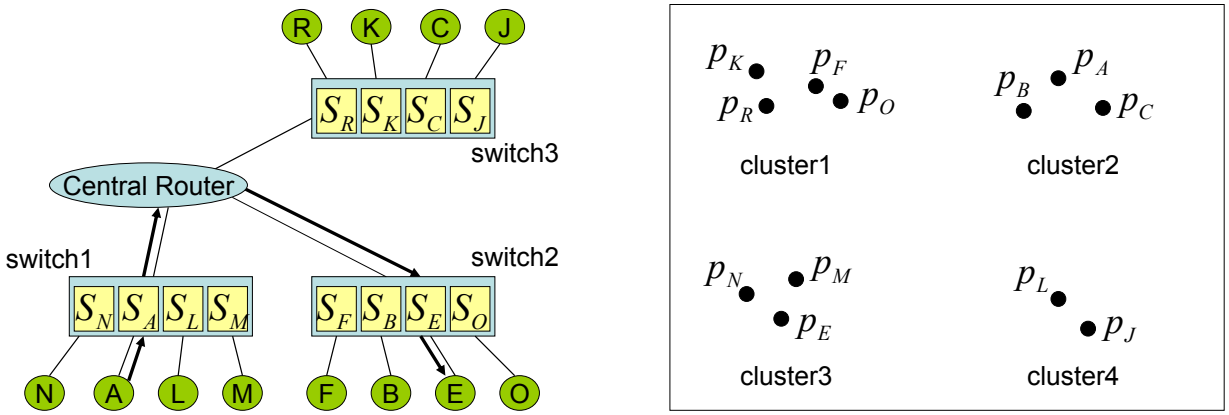
Similarly, when traffic arrives to *switch2*, the cluster-based AD sensor of host *E* (S_E), checks it not only against its own behavior profile p_E but also against the profiles in the cluster where *E* is a member i.e., *cluster3* that contains the behavior profiles p_M and p_N. Equation 3 summarizes the decision making process that takes place on S_E,

$$S_E = \begin{cases} 1 & \text{if } \forall p_i \in cluster3 \;\; D_{p_i}(t) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

3.1 Behavior Profiles and Clusters Update

Over time, network hosts may change their behavior thus requiring the computation of new profiles as well as the update of the cluster distribution. Whenever new behavior profiles are computed by the cluster-based AD sensors, these are sent to the central router which updates the cluster distribution. Once updated, the new cluster distribution is broadcasted to all the switches in the network. The update of each behavior profile is not performed continuously, instead each profile is trained by *epochs* whenever major changes in the behavior of the host occur. In Section 5 we evaluate the frequency of profile update and its network bandwidth requirements.

It is important to clarify that by locating the profile computation and the anomaly detection on the switch, we eliminate the possibility of a host lying about the existence of an alert. However, if the computation of the behavior profiles



(a) Schematics of a network with multiple switches and hosts (*circles*). Each box within the switch represents a cluster-based AD sensor for each of the hosts in the subnet. The arrows indicate a traffic exchange from host *A* to host *E*.

(b) Clusters of behavior profiles determined by the clustering algorithm. In this instance, four different clusters of behavior profiles have been identified by the central router among all hosts in the network.

Figure 3. Cluster-based Anomaly Detection for a traffic exchange between hosts A and E.

is performed at the hosts rather than at the switch, it is possible that the host itself fabricates a malicious profile that deems anomalous traffic as normal. To guard against this possibility, the central router must be enhanced by incorporating an algorithm such as the one presented in [5]. Prior to computing the new cluster distribution, the algorithm would allow the central router to differentiate between new profiles that differ slightly from previous profiles (*concept drift*) and hosts trying to maliciously craft their profiles (*attacks*).

4 Experimental Results and Comparative Analysis

In this section, we present an actual evaluation of cluster-based AD sensors based on real behavior profiles from wireless users. We also contrast these results with similar experiments performed with single-profile AD sensors. For this purpose, we proceeded to compute behavior profiles of wireless users in a real network. We used tcpdump data containing packet headers of wireless traffic captured from users at Dartmouth College. Specifically, we employed one month of traffic from the Fall03 tcpdump dataset (163GB) in the *CRAWDAD* repository [2]. This month of traffic was assumed to be clean of attacks and was thus considered *ground truth*. We focused our analysis on four distinct ports (services): port 21 (ftp), port 22 (ssh), port 25 (smtp), and port 80 (http). For each port, we randomly identified 100 different MAC addresses that exhibited output traffic to the service. Each MAC address was assumed to represent a different user in the network. Moreover, we associated each user with a unique host in the network i.e., there were no hosts with multiple users. Hereafter, we use the terms host and user interchangeably.

We computed the behavior profile of each of these users on a per-port basis adopting the first week of the month as the training set. To be able to detect different types of network-based attacks, both control and data packets were considered during training. We defined the behavior profile p_i (see Equation 4) of user i as a set of hourly histograms h_{f_n} for each feature f_n, where f_n represents a measure of network-related statistics. Specifically, the following features were modeled: average number of unique users contacted per hour, average number of packets exchanged per hour, and average length of the packets exchanged per hour. Each histogram h_{f_n}, computed per port (service) and direction (input or output), represents the hourly average and standard deviation for a feature f_n. The hourly average a_j and hourly standard deviation σ_j were measured each hour j of the day and averaged throughout the duration of the training period. Hence,

$$p_i = \{h_{f_1}, ..., h_{f_n}\} \qquad (4)$$
$$h_{f_n} = \{(a_0, \sigma_0), (a_1, \sigma_1), ..., (a_{23}, \sigma_{23})\}$$

The performance of the single-profile and cluster-based AD sensors was quantified using two parameters: the *false positive rate (FP)* and the *detection threshold* (Φ). The *FP* measures the fraction of normal traffic that has been erroneously considered anomalous over all the testing traffic. In the case of a single-profile AD sensor, the hourly traffic t of a user was considered anomalous when it deviated more than one standard deviation σ_j from its own behavior profile hourly average a_j. In contrast, the cluster-based AD sensor checked that the hourly traffic t was within one standard

deviation σ_j of its own profile hourly average a_j as well as within one standard deviation of the hourly averages of all the members of its own cluster (see Equation 1).

Due to the fact that the Dartmouth wireless traffic did not contain any worm traces, a direct measurement of the detection rate (anomalous traffic detected as such) was not feasible. Instead, we defined a new parameter, Φ_i, that represents the detection threshold used by the AD sensor of user i to determine whether certain traffic was anomalous. We hypothesize that the detection threshold Φ_i of a sensor is correlated to its detection rate and thus provides an indirect measurement of its trend. For the single-profile AD sensor, we computed the detection threshold $\Phi_i(j)$ per hour j for user i by adding the standard deviation to the average value observed during training for a specific hour i.e., $\Phi_i(j) = a_j + \sigma_j$. On the other hand, the value of the detection threshold $\Phi_i(j)$ for the cluster-based AD sensor was computed by determining the maximum value among the detection thresholds $\Phi_m(j)$ of all the profiles in the cluster c where i is a member. Hence,

$$\Phi_i(j) = \quad \forall m \in c \;\; MAX(\Phi_m(j)) \tag{5}$$

This is consistent with the assumption that traffic is considered anomalous only when all profiles in a cluster agree. As a result, the cluster detection threshold $\Phi_i(j)$ corresponds to the maximum value among all cluster members. Finally, the average value Φ_i for each single-profile and cluster-based AD sensor was computed by averaging the 24 hourly detection thresholds $\Phi_i(j)$.

For each of the four ports, the 100 user behavior profiles were clustered with the *k-means* algorithm [5]. Armed with these clusters of behavior profiles, we assigned a cluster-based AD sensor to each of the 100 selected users. Using the second week of the *CRAWDAD* tcpdump data as the testing set, we proceeded to compute the FP and Φ_i of each cluster-based AD sensor. The resulting values were then averaged to produce the average FP ($\overline{FP}$) and the average Φ_i ($\overline{\Phi}$) across all users. For comparison purposes, we repeated the same tests assigning a single-profile AD sensor to each of the 100 users.

Figure 4 summarizes the $\overline{FP}$ for the cluster-based and single-profile AD sensors calculated for ports 21, 22, 25, and 80. The results show the $\overline{FP}$ when the behavior profiles of the sensors are modeled with the following features: hourly average number of unique users contacted (*i*), hourly average number of packets exchanged (*ii*), hourly average length of the packets exchanged (*iii*), or a combination of the three features (*iv*). As can be seen from Figure 4, the $\overline{FP}$ of cluster-based AD sensors is lower than that of single-profile AD sensors across different features and ports. In all cases, the cluster-based $\overline{FP}$ rate is at least halved when compared to its single-profile counterpart. On

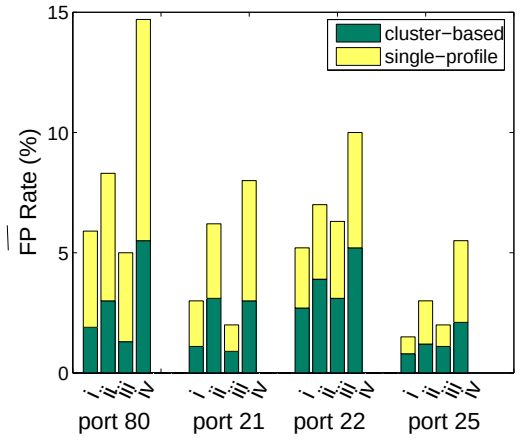


Figure 4. $\overline{FP}$ rates for single-profile and cluster-based AD sensors computed for ports 80, 21, 22, and 25. Average rates per port are reported for profiles modeled with the following features: number of unique users contacted (*i*), number of packets exchanged (*ii*), length of the packets exchanged (*iii*) and a combination of these three features (*iv*). As can be seen, the $\overline{FP}$ rate of cluster-based AD sensors is at least halved when compared to single-profile AD sensors.

the other hand, Figure 5 depicts the ratio of cluster-based $\overline{\Phi}$ ($\overline{\Phi}_{cluster}$) to single-profile $\overline{\Phi}$ ($\overline{\Phi}_{single}$) for ports 21, 22, 25, and 80. Ratios per port are reported for profiles modeled with the following features: number of unique users contacted (*i*), number of packets exchanged (*ii*), and length of the packets exchanged (*iii*). The ratio for the combination of these three features (*iv*) corresponds to the set of individual ratios for each feature and is thus omitted from the Figure. In this instance, we chose to display ratios rather than the individual values of $\overline{\Phi}$ in order to normalize the results to the same scale. The ratios shown in Figure 5 indicate that the cluster-based $\overline{\Phi}$ is increased by at most 1/6 of its single-profile counterpart across all features and ports. The main conclusion we draw from these results is that on average a cluster-based AD sensor can significantly decrease the FP rate while slightly increasing Φ_i with respect to a single-profile AD sensor. While it may be argued that a single-profile AD sensor could potentially be tuned to accomplish similar performances as the cluster-based AD sensor, we believe the latter approach still enhances single-profile AD sensors by providing a way to automatically tune FP and Φ_i guided by the boundaries imposed by the behavior profiles in the cluster.

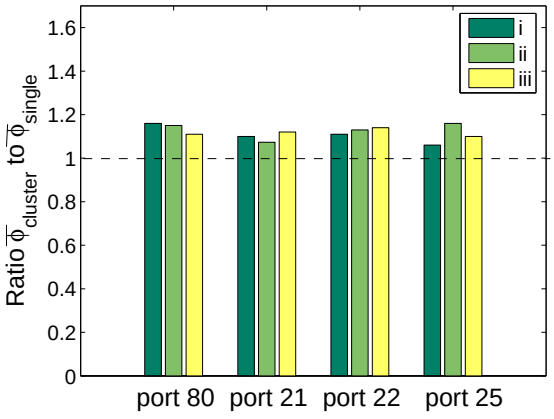


Figure 5. Ratio of cluster-based $\overline{\Phi}$ ($\overline{\Phi}_{cluster}$) to single-profile $\overline{\Phi}$ ($\overline{\Phi}_{single}$) computed for ports 80, 21, 22, and 25. Ratios per port are reported for profiles modeled with the following features: number of unique users contacted (*i*), number of packets exchanged (*ii*), and length of the packets exchanged (*iii*). The ratio for the combination of these three features (*iv*) corresponds to the set of individual ratios for each feature and is thus omitted from the Figure. Ratios are above unity which we interpret as a slight increment in the detection threshold of each sensor (Φ_i).

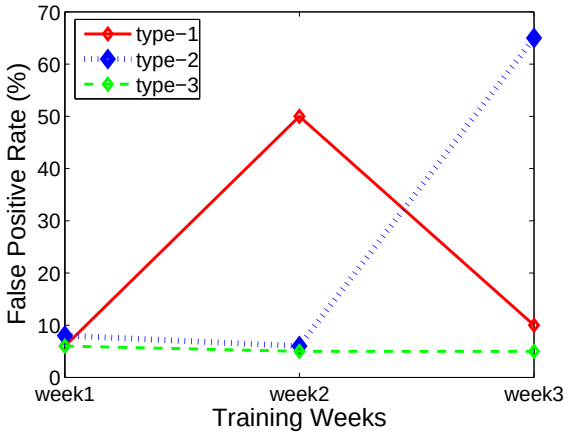


Figure 6. Evolution of False Positive Rates (%) for different training lengths of wireless users in the *CRAWDAD* dataset.

The decrease in the cluster-based FP rates is possibly related to the fact that most false alerts correspond to tiny fluctuations in the average value of the features, which are greatly reduced when compared to the profile of other users with similar behavior. These tiny fluctuations may be related to the high variability of wireless networks in terms of packet fragmentation or packet redundancy, which prevents the acquisition of a robust behavior profile. Although one may intuitively think that training the behavior profile for longer periods would decrease the FP rates in single-profile AD sensors, behavior profiles of wireless users change frequently in relatively short periods of time. Thus, the identification of clusters of users that share similar behavior provides a broader view of *normal behavior* in short training periods and enhances the robustness of the anomaly detection by decreasing the volume of false alerts.

5 Network Bandwidth Requirements

Behavior profiles are computed as a set of histograms, where each histogram represents a specific feature and is composed of 24 hourly averages. The simplest profile consisting of a single histogram has an approximate size of 97 bytes. More complex profiles similar to the ones used in our experiments with three histograms (three features) for four different ports can reach a size of 1164bytes. Therefore, the communication of a profile to the central router in our experimental setup implies a bandwidth use of at most $\sim$ 1KByte. Every time the central router receives an updated profile, it proceeds to recompute the clusters of behavior profiles and to broadcast the new configuration to all the switches in the network. Assuming a network of 10,000 machines, the broadcast to all the switches would amount to a transmission between 970KBytes and 11.6MBytes.

A transmission between 970KBytes and 11.6MBytes constitutes an acceptable bandwidth use for a one-time transaction. However, because this transmission is to be performed every time a behavior profile is updated, it is also important to estimate the frequency of the profile updates which will then provide an actual measure of the total bandwidth requirements. In order to estimate this frequency, we randomly selected a group of users from the same tcpdump file as in our experiments. Each individual user was trained and tested for three different lengths of time: 1) training for the first week of data and computing the FP for the second week of data, 2) training for the first two weeks of data and computing the FP for the third week of data, and 3) training for the first three weeks of data and computing the FP for the fourth week of data. For simplicity, tests were only conducted using the number of packets exchanged on port 80, which displayed the highest amount of traffic and variability.

Figure 6 shows the evolution of the FP rate for three distinct users (*type-1*, *type-2*, and *type-3*) representative of

common patterns among other users in the dataset. Users similar to *type-1* displayed low FP rates following the first week of training, which can be interpreted as little variability in behavior between the first and the second week. After two weeks of training, users similar to *type-1* experienced a peak in the FP rate. Examining the behavior profiles, we noticed that this FP peak was related to a sudden increase in the hourly average values of the number of packets exchanged. After three weeks of training, the FP rate decreased to levels comparable to the first week. Users similar to *type-2* displayed low FP rates after the first and second weeks of training. A peak in the FP rate was reached after three weeks of training that was also associated to an increase in the number of packets exchanged. Finally, users similar to *type-3* displayed a slightly decreasing FP rate over the three different training lengths.

This analysis shows that wireless users are very dynamic and as a result their behavior profiles change frequently on scales as short as one week. If all the users in the network synchronize the beginning of their training periods, the broadcast of the new configuration of clusters of behavior profiles would need to be done on a weekly timescale e.g., on a weekend night. Therefore, the possibility of network saturation is greatly reduced. Clearly, our analysis does not account for every single user behavior in the network, but it shows distinct behavioral patterns shared by a significant fraction of users.

6 Related Work

6.1 False Alert Reduction through Redundancy

Redundancy has been widely studied in the literature as a means to enhance the security of a system [6], [9]. The *Cooperative Security Managers* (CSM) approach describes a fully distributed intrusion detection system without any central coordinator [16]. In this approach, each intrusion detection (ID) sensor located on a host focuses on detecting anomalies in the behavior of its local users. Furthermore, each ID sensor may model the behavior of users from other hosts and confront any alert with the original host. The AAFID architecture uses redundancy as a means to enhance the performance of anomaly detection sensors [1], [13]. AAFID employs multiple agents that work in parallel within a host to detect anomalous behavior. Final decisions on the nature of the behavior are reached only when different host agents agree.

The main problem associated with redundancy is its tendency to overload the host since the ID sensor computations take away computing cycles from real host applications. In our approach, a *form of redundancy* is achieved through the use of clusters of behavior profiles. Specifically, the AD sensor of each host is enhanced with behavior profiles of hosts with similar behavior. Through these clusters of behavior, the sensor manages to gather a *form of redundant information* without having to compute multiple profiles but rather by exchanging behavior profiles with other hosts. This enhancement reduces the volume of false alerts while keeping the computational load balanced.

6.2 False Alert Reduction through Collaboration

EMERALD is amongst the first architectures to incorporate the use of collaborative sensors to correlate alerts within one network or across different domains for anomaly detection [12]. The use of correlation algorithms aids in the detection of coordinated attacks while decreasing the volume of false alerts. GrIDS collects network activity from various locations within the network and builds graphs that help discover large-scale coordinated attacks [14]. WORMINATOR is a system that exchanges alerts based on anomalous content detected by AD sensors installed in hosts across different institutions [10], [11]. DShield is a community-based collaborative log-correlation system [15].

These collaborative approaches can work effectively as long as there exists a coordinated attack on different hosts in the network. However, alert sharing is less effective in reducing the volume of false alerts during isolated attacks on a single host since other sensors in the network do not observe the attack. In our approach, we introduce collaboration through profile sharing in the form of cluster of behavior profiles. By sharing behavior profiles rather than alerts, the AD sensor can reduce the volume of false alerts in isolated attacks. This follows from the fact that the attacked host stores other similar host behavior profiles and can predict their decision had they observed the attack.

6.3 Deployment of AD Sensors

AD sensors can been deployed in different types of architectures [9]. The CSM architecture is a *fully distributed* architecture that locates an ID sensor on each of the hosts in the network [16]. The hosts are then responsible for their own anomaly detection as well as for correlating alerts coming from other ID sensors located in other hosts. The NADIR architecture uses a *hierarchical* approach where *service nodes* located at Los Alamos National Laboratory's Integrated Computed Network (ICN) are responsible for the anomaly detection while the alert analysis is performed at a central expert system [7]. The AAFID architecture also employs a *hierarchical* approach where each host has multiple sensor agents that generate different types of alerts [1]. These alerts are then processed at higher level *monitors*.

A weakness in locating the anomaly detection on the individual hosts is that it potentially allows them to fabricate false alerts. In our architecture, the switches are responsible for computing the behavior profile of the hosts in its subnet as well as for the anomaly detection. Thus, the possibility of a host maliciously crafting an alert or lying about its own profile is eliminated.

7 Conclusions and Future Work

We have presented a cluster-based AD sensor that reduces the volume of false alerts generated by single-profile AD sensors. The advantage of this approach is that it uses clusters of behavior profiles to provide a broader definition of normal behavior that compensates for poor or insufficient training data commonly observed in single-profile AD sensors. We have also introduced an architecture design that maximizes the efficiency of cluster-based AD sensors. In our architecture, the anomaly detection is performed on the switches rather than on the hosts, thus eliminating the possibility of false alerts maliciously crafted by the hosts. We have experimentally shown that for real wireless users in the *CRAWDAD* dataset the volume of false alerts using cluster-based AD sensors is at least halved when compared to single-profile AD sensors.

Future work will focus on evaluating cluster-based AD sensors using profiles that model additional features related to specific network attacks e.g., number of SYN/ACK packets to detect SYN flood attacks. We also plan to analyze more simplistic temporal models other than histograms such as daily and hourly averages. Lastly, we will evaluate the impact of shorter and longer profile training periods on the performance of cluster-based AD sensors.

The work presented thus far has relied on behavior profiles that model the characteristics of the network traffic exchanged by a host. In the future, we plan to extend our work to behavior profiles computed based on user or application characteristics at the hosts e.g., commands executed by a user or the interaction between an application and the operating system. The main goal will be to understand whether profiles based on user or application behavior can be used effectively by cluster-based AD sensors to reduce the volume of false alerts.

Acknowledgements

This work was partially supported by NSF Grant CNS-06-27473 and by DARPA Grant HR0011-06-1-0034. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or the U.S. Government. We would like to thank Stefano Pacifico for his work on the CRAWDAD dataset.

References

[1] J. Balasubramaniyan et al. *An Architecture for Intrusion Detection using Autonomous Agents,* ACSAC, 1998.

[2] Crawdad Dataset, Dartmouth University. *http://crawdad.cs.dartmouth.edu/,* 2004.

[3] D. E. Dennning. *An Intrusion Detection Model,* IEEE Trans. on Soft. Eng., 1987.

[4] V. Frias-Martinez and S. J. Stolfo and A. D. Keromytis. *Behavior-Based Access Control: A Proof-of-Concept,* Information Security Conference (ISC), 2008.

[5] V. Frias-Martinez and J. Sherrick and S. J. Stolfo and A. D. Keromytis. *A Network Access Control Mechanism Based on Behavior Profiles,* Technical Report cs-001-09.

[6] M. A. Hiltunen and R. D. Schlichting and C. A. Ugarte. *Enhancing Survivability of Security Systems Using Redundancy,* DSN, 2001.

[7] J. Hochberg and K. Jackson and C. Stallings et al. *NADIR: An automated systems for detecting network intrusion and misuse,* Computers and Security, 1993.

[8] V. Kumar and J. Srivastava and A. Lazarevic. *Managing Cyber Threats: Issues, Approaches and Challenges,* Springer Publisher, 2005.

[9] B. Littlewood and L. Strigini. *Redundancy and Diversity in Security,* European Symposium on Research in Computer Security (ESORICS), 2004.

[10] M. E. Locasto and J. J. Parekh et al. *Towards Collaborative Security and P2P Intrusion Detection,* IEEE Workshop on Information Assurance and Security, 2005.

[11] J .J. Parekh and K. Wang and S. J. Stolfo. *Privacy-Preserving Payload-Based Correlation for Accurate Malicious Traffic Detection,* LSAD, 2006.

[12] P. A. Porras and P. G. Neumann. *Event Monitoring Enabling Responses to Anomalous Live Disturbances,* NISS, 1997.

[13] E. H. Spafford and D. Zamboni. *Intrusion detection using autonomous agents,* Computer Networks, 2000.

[14] S. Staniford-Chen et al. *GrIDS – (A) Graph-based Intrusion Detection System for Large Networks,* NISS, 1996.

[15] J. Ullrich *DShield Homepage www.dshield.org*

[16] G. B. White and E. A. Fisch and U. W. Pooch. *Co-operating security managers: A peer-based intrusion detection system,* IEEE Network, January/February 1996.