ISSN:2231-0711

K.Tejaswi et al | IJCSET |May 2013 | Vol 3, Issue 5, 177-182

A Review of Online Rogue Access Point Detection

K.Tejaswi, Dr.Syed Umar, K.Bhavana

Department of ECM,

KL University, A.P., INDIA

***Abstract*— A rogue access point is a wireless access point that has either been installed on a secure company network without explicit authorization from a local network administrator or has been created to allow a hacker to conduct a man-in-the-middle attack. In this paper, we propose two online algorithms to detect rogue access points using sequential hypothesis tests applied to packet-header data collected passively at a monitoring point. One algorithm requires training sets, while the other does not. Both algorithms extend our earlier TCP ACK-pair technique to differentiate wired and wireless LAN TCP traffic, and exploit the fundamental properties of the 802.11 CSMA/CA MAC protocol and the half duplex nature of wireless channels.**

***Keywords*—Rogue access point detection, Sequential hypothesis testing, TCP ACK-pairs.**

I. INTRODUCTION

The deployment of IEEE 802.11 wireless networks (WLANs) has been growing at a remarkable rate during the past several years. The presence of a wireless infrastructure within a network, however, raises various network management and security issues. One of the most challenging issues is rogue access points (APs), i.e., wireless access points that are installed without explicit authorization from a local network management [9, 1, 3, 4]. Although usually installed by innocent users for convenience or higher productivity, rogue APs pose serious security threats to a secured network. They potentially open up the network to unauthorized parties, who may utilize the resources of the network, steal sensitive information or even launch attacks to the network. Furthermore, rogue APs may interfere with nearby well-planned APs and lead to performance problems inside the network. Due to the above security and performance threats, detecting rouge APs is one of the most important tasks for a network manager. Broadly speaking, two approaches can be used to detect rogue APs. The first approach detects rogue APs by monitoring the RF airwaves, possibly exploiting additional information gathered at routers and switches [2, 8, 9, 1, 3, 4, 10, 11, 27]. The second approach monitors incoming traffic at a traffic aggregation point (e.g., a gateway router) and determines whether a host uses wired or wireless connection1. If a host is determined as using wireless connection while it is not authorized to do so (e.g., it is not contained in the authorization list), the AP attached by this host is detected as a rogue AP. The first approach can suffer from various drawbacks including scalability, deployment cost, effectiveness and accuracy (see Section 1.1). The second approach does not have the above drawbacks: (1) since it is based on passive measurements at a single monitoring point, it is scalable, requiring little deployment cost and effort, and is easy to manage and maintain; (2) since the detection is by detecting wireless connections, it is equally applicable to detect layer-2 or layer-3 rogue devices while the first approach may need different schemes for rogues at different layers [10, 11]. The challenge in applying the second approach is: how to effectively detect wireless traffic from passively collected data in an online manner? In this paper, we take the second approach and develop two online algorithms to meet the above challenges. Our main contributions are as follows:

• We extend the analysis in [25] and demonstrate that using TCP ACK-pairs can effectively differentiate Ethernet and wireless connections (including both 802.11b and 802.11g).

• We develop two online algorithms to detect rogue APs. Both algorithms use sequential hypothesis tests and make prompt decisions as TCP ACK-pairs are observed. One algorithm requires training data, while the other does not. To the best of our knowledge, ours are the first set of passive online techniques that detect rogue APs by differentiating connection types.

• We have built a system for online rogue-AP detection using the above algorithms and deployed it at the gateway router of the University of Massachusetts, Amherst (UMass). Extensive experiments in various scenarios have demonstrated the excellent performance of our algorithms:

(1) The algorithm that requires training provides rapid detections and is extremely accurate (the detection is mostly within 10 seconds, with very low false positive and false negative ratios);

(2)The algorithm that does not require training detects 60%-76% of the wireless hosts without any false positives;

(3) Both algorithms are light-weight, with computation and storage overhead well within the capability of commodity equipment. We further conduct experiments to demonstrate that our scheme can detect connection-type switching and wireless networks behind a NAT box, and it is effective even when the hosts have high CPU, disk or network utilizations.

II. RELATED WORK

As mentioned earlier, monitoring RF waves and IP traffic are two broad classes of approaches to detecting rogue APs. Most existing commercial products take the first approach they either manually scan the RF waves using sniffers (e.g., AirMagnet [2], NetStumbler [8]) or automate the process using sensors (e.g.,[1, 9, 4]. Automatic scanning using sensors is less time consuming than manual scanning and provides a continuous vigilance to rogue APs. However, it may require a large number of sensors for good coverage, which leads to a high deployment cost. Furthermore, since it depends on signatures of APs (e.g., MAC address, SSID, etc.), it becomes ineffective when a rogue AP spoofs signatures. Three recent research efforts [10, 11, 27] also

use RF sensing to detect rogue APs. In [10], wireless clients are instrumented to collect information about nearby APs and send the information to a centralized server for rogue AP detection. This approach is not resilient to spoofing. Secondly, it assumes that rogue APs use standard beacon messages in IEEE 802.11 and respond to probes from the clients, which may not hold in practice. Last, all unknown APs (including those in the vicinity networks) are flagged as rogue APs, which may lead to a large number of false positives. The main idea of [11] is to enable dense RF monitoring through wireless devices attached to desktop machines. This study improves upon [10] by providing more accurate and comprehensive rogue AP detection. However, it relies on proper operation of a large number of wireless devices, which can be difficult to manage. In contrast, our approach only requires a single monitoring point, and is easy to manage and maintain. The focus of [27] is on detecting protected layer3 rogue APs. Our approach is equally applicable to detect layer-2 or layer-3 rogue devices. The studies of [13, 19] detect rogue APs by monitoring IP traffic. The authors of [13] demonstrated from experiments in a local test bed that wired and wireless connections can be separated by visually inspecting the timing in the packet traces of traffic generated by the clients. The settings of their experiments are very restrictive. Furthermore, the visual inspection method cannot be carried out automatically. Our schemes are based on a rigorous analysis of Ethernet and wireless traffic characteristics in realistic settings. Furthermore, we provide two sequential hypothesis tests to automatically detect rogue APs in real time. The technique in [19] requires segmenting large packets into smaller ones, and hence is not a passive approach. There are several prior studies on determining connection types. However, none of them provides a passive online technique, required for our scenario. Our previous work [25] proposes an iterative Bayesian inference technique to identify wireless traffic based on passive measurements. This iterative approach is not suitable for online deployment. The work of [12] uses entropies to detect wireless connection in an offline manner. In other studies, differentiating connection types is based on active measurements [26] or certain assumptions about wireless links (such as very low bandwidth and high loss rates) [15], which do not apply to our scenario.
Last, sequential hypothesis testing provides an opportunity to make decisions as data come in, and thus is a suitable technique for our purpose. It is also used for prompt port scan detection.

III. Problem Setting And Approach

Consider a local network (e.g., a university campus or an enterprise network), as illustrated in Fig. 1. A monitoring point is placed at an aggregation point (e.g., the gateway router) of this local network, capturing traffic coming in and going out of the network. End hosts within this network use either wired Ethernet or 802.11 WLAN to access the Internet. An end host not authorized to use WLAN may install a rogue AP to connect to the network. Our goal is to detect those rogue APs in real time based on passive measurements at the monitoring point. For this purpose, we must answer the following two questions: (1) what statistics can be used to effectively detect wireless hosts? (2) how to detect wireless hosts in an online manner? We next provide a high-level description on how we address these two questions; a detailed description is deferred to Sections 3 and 4.We have shown that inter-ACK time is a statistic that can be used to effectively detect wireless hosts in [25]. An inter-ACK time is the inter-arrival time of a TCP ACK-pair, i.e., a pair of ACKs corresponding to two data packets that arrive

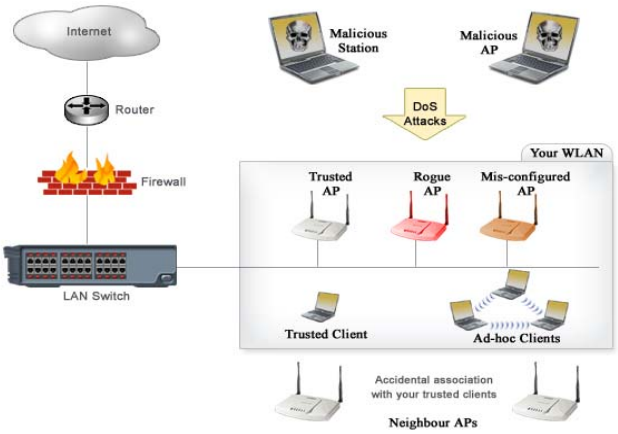

Figure 1: Problem setting: a monitoring point at an aggregation point captures incoming traffic and outgoing traffic to detect rogue APs.

at the monitoring point close in time. In [25], we analyze the inter-ACK time in Ethernet and WLAN and demonstrate that it can be used to differentiate these two connection types. However, the analysis does not include 802.11g, since it was not widely deployed at that time. In Section 3, we extend the analysis in [25] to 802.11g, and derive a new set of results for Ethernet and 802.11b. Our results demonstrate that inter-ACK times can effectively differentiate Ethernet and WLAN (including both 802.11b and 802.11g hosts). For online detection of wireless hosts, we develop two light-weight algorithms (see Section 4), both using sequential hypothesis tests and taking the inter-ACK times as input. These two algorithms roughly work as follows. They calculate the likelihoods that a host uses WLAN and Ethernet as TCP ACK-pairs are observed. When the ratio of the WLAN likelihood against the Ethernet likelihood exceeds a certain threshold, they make a decision that the host uses WLAN.

IV. Online Detection Algorithms

In this section, we develop two online algorithms to detect wireless hosts based on our analysis in the previous section. Both algorithms use sequential hypothesis test technique and take the inter-ACK times as the input. The first algorithm requires knowing the inter-ACK time distributions for Ethernet and WLAN traffic a priori. The second algorithm does not have such a requirement. Instead, it is directly based on Theorems 1 and 2 (see Section 3). We refer to these two algorithms as sequential hypothesis test with training and sequential hypothesis test without training respectively. The algorithm without training, although is not as powerful as the one with training (see Section 7), is suitable for scenarios where the inter-ACK time distributions are not available a priori (e.g., for organizations with no wireless networks). We now

describe these two algorithms in detail. Both algorithms use at most N = 100 ACK-pairs to make a decision (i.e., whether the connection is Ethernet or WLAN) to accommodate the scenarios where a host switches between Ethernet and WLAN connections.

IV.I Sequential Hypothesis Test with Training

We have demonstrated that the inter-ACK time distributions for Ethernet and WLAN differ significantly (see Section 3). When these distributions are known, we can calculate the likelihoods that a host uses Ethernet and WLAN respectively given a sequence of observed inter-ACK times. If the likelihood of using WLAN is much higher than that of using Ethernet, we conclude that the host uses WLAN (and vice versa). We now describe the test in more detail. Let {δA i }n i=1 represent a sequence of inter-ACK time observations from a host, and {_Ai }n i=1 represent their corresponding random
variables. Let E and W represent respectively the events that a host uses Ethernet and WLAN. Let LE = P(_A1 = δA1 ,_A2 = δA2 , . . . ,_An = δA n | E) be the likelihood that this observation sequence is from an Ethernet host. Similarly, let LW = P(_A1 = δA 1 ,_A2 = δA 2 , . . . ,_An = δA n | W) be the likelihood that the observation sequence is from a WLAN host. Let pi = P(_Ai = δA i | E) be the probability that the i-th inter-ACK time has value δA i given that it is from an Ethernet host. Similarly, let qi = P(_Ai = δA i | W) be the probability that the i-th inter-ACK time has value δA i given that it is from a WLAN host. Both pi and qi are known obtained from the inter-ACK time distributions for Ethernet and WLAN traffic respectively. Assuming that the inter ACK times are independent and identically distributed, we have

$$L_E = P(\Delta_1^A = \delta_1^A, \ldots, \Delta_n^A = \delta_n^A \mid E) = \prod_{i=1}^{n} p_i,$$

$$L_W = P(\Delta_1^A = \delta_1^A, \ldots, \Delta_n^A = \delta_n^A \mid W) = \prod_{i=1}^{n} q_i.$$

This test updates LW and LE as an ACK-pair is observed. Let K > 1 be a threshold. If after the n-th ACK-pair, the ratio of LW and LE is over the threshold, i.e., LW/LE > K, then the host is classified as a WLAN host. If LW/LE < 1/K, then the host is classified as an Ethernet host.

$n = 0, l_E = l_W = 0.$
do {
  Identify an ACK-pair
  $n = n + 1$
  $p_n = P(\Delta_n^A = \delta_n^A \mid E), q_n = P(\Delta_n^A = \delta_n^A \mid W)$
  $l_E = l_E + \log p_n$, $l_W = l_W + \log q_n$

  if $l_W - l_E > \log K$
    Report WLAN. $n = 0, l_E = l_W = 0.$

  else if $l_W - l_E < -\log K$
    Report Ethernet. $n = 0, l_E = l_W = 0.$

  else if $n = N$
    Report undetermined. $n = 0, l_E = l_W = 0.$
}

Figure 3: Sequential hypothesis test with training N = 100.

If neither decision is made after N ACK-pairs, the connection type is classified as undetermined. In the implementation, for convenience, we use log-likelihood function lw = log(LW) and lE = log(LE) instead of the likelihood function. This test is summarized in Fig. 3. As we can see, it has very little computation and storage overhead (it only stores the current likelihoods for Ethernet and WLAN for each IP address being monitored).

IV.II Sequential Hypothesis Test without Training

This test does not require knowing the inter-ACK time distributions for Ethernet and WLAN hosts a priori. Instead, it leverages the analytical results that the probability of an inter-ACK time exceeding 600 μs is small for Ethernet hosts, while it is much larger for WLAN hosts. In the following, we first construct a likelihood ratio test [14], and then derive from it a sequential hypothesis test. The likelihood ratio test is as follows. Let p be the probability that an inter-ACK time exceeds 600 μs, that is, p = P(_A > 600 μs). By Theorem 1, we have p < θ = 0.18 for Ethernet host. Therefore, if the hypothesis p < θ is rejected by the inter-ACK time observation sequence, we conclude that this host does not use Ethernet and hence uses WLAN. More specifically, consider two hypotheses,H0 and

$m = n = 0.$
do {
  Identify an ACK-pair
  $n = n + 1$
  $m = m + \mathbf{1}(\delta_n^A \geq 600\ \mu s)$
  $\hat{p} = \frac{m}{n}$

  if $\hat{p} = 1$ **and** $n > -\frac{\log K}{\log \theta}$
    Report WLAN. $m = n = 0.$

  else if $n < \frac{m(\log \hat{p} - \log \theta + \log(1-\theta) - \log(1-\hat{p})) - \log K}{\log(1-\theta) - \log(1-\hat{p})}$
    Report WLAN. $m = n = 0.$

  else if $n \geq 43$ **and** $\hat{p} \geq 0.5$
    Report WLAN. $m = n = 0.$

  else if $n = N$
    Report undetermined. $m = n = 0.$
}

Figure 4: Sequential hypothesis test without training, , where 1(•) is the indicator function, N = 100.

Ha, representing respectively the null hypothesis that a host uses Ethernet and the alternative hypothesis that the host uses WLAN. For a sequence of inter-ACK time observations {δA
i }n i=1, let m be the number of observations that exceed 600 μs. Let K > 1 be a threshold. Then the likelihood ratio test rejects the null hypothesis H0 when

$$\lambda = \frac{\sup_{0 \leq p \leq \theta} p^m (1-p)^{n-m}}{\sup_{0 \leq p \leq 1} p^m (1-p)^{n-m}} < \frac{1}{K}$$

In the middle term above, the numerator is the maximum probability of having the observed sequence (which has m inter-ACK times exceeding 600 μs) computed over parameters in the null hypothesis (i.e., 0 ≤ p ≤ θ). The denominator of λ is the maximum probability of having the observed sequence over all possible parameters (i.e., 0 ≤ p ≤ 1). If λ < 1/K, that is, there are parameter points in the alternative hypothesis for which the observed sample is much more likely than for any parameter points in the null hypothesis, the likelihood ratio test concludes that H0 should be rejected. In other words, if λ < 1/K, the likelihood ratio test concludes that the host uses WLAN. We now derive a sequential hypothesis test from the above likelihood ratio test. Let ˆp = m/n, where m is the number of inter-ACK times exceeding 600 μs and n is the total number

of inter-ACK times. It is straightforward to show that ˆp is the maximum likelihood estimator of p, i.e., sup0≤p≤1 pm(1−p)n−m is achieved when p = ˆp. When ˆp ≤ θ, we have sup0≤p≤_ pm(1 −p)n−m = sup0≤p≤1 pm(1− p)n−m, and hence λ = 1 > 1/K. In this case, the null hypothesis H0 is not rejected. Therefore, we only consider the case where θ < ˆp, which can be classified into two cases:

Case 1: θ < ˆp < 1. In this case, to reject the null hypothesis H0, we need

$$\frac{\hat{p}^m(1-\hat{p})^{n-m}}{\theta^m(1-\theta)^{n-m}} > K$$

which is equivalent to

$$n < \frac{m(\log\hat{p} - \log\theta + \log(1-\theta) - \log(1-\hat{p})) - \log K}{\log(1-\theta) - \log(1-\hat{p})}. \quad (1)$$

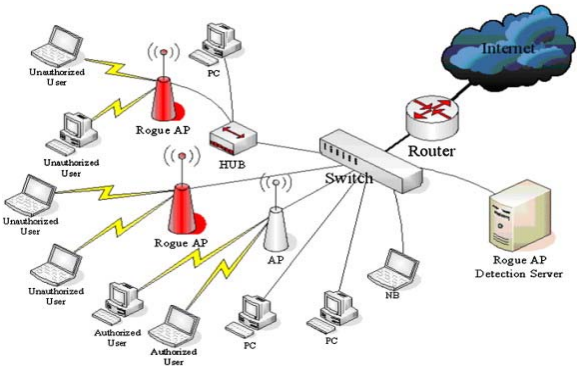

Figure 5: Online rogue-AP detection system.

Case 2: ˆp = 1. In this case, to reject the null hypothesis H0, we need

$$\frac{1}{\theta^n} > K$$

which is equivalent to

$$n > -\frac{\log K}{\log\theta}. \quad (2)$$

When K = 106 and θ = 0.18, from (2), we have n ≥ 8. This implies that we need at least 8 ACK-pairs to detect a WLAN host for the above setting. In addition to conditions (1) and (2), we also derive a complementary condition to reject the null hypothesis H0 directly from Theorem 2. Theorem 2 states that, when the number of inter-ACK observations n is between 43 and 100, we have P(ξn .5(_A) ≤ 600 μs) ≈ 1 for Ethernet hosts. Therefore, an additional condition to reject H0 is when 43 ≤ n ≤ 100 and ˆp > 0.5 (because this condition implies that at least half of the inter-ACK observations exceed 600 μs, that
is, ξn .5(_A) > 600 μs, which contradicts Theorem 2). We combine the above three conditions to construct a sequential hypothesis test as shown in Fig. 4. As we can see, this test has very little computational and storage overhead (it only stores the total number of inter-ACK times and the number of inter-ACK times exceeding 600 μs for each IP address being monitored). Last, note that it only reports WLAN hosts, while the sequential hypothesis test with training reports both WLAN and Ethernet hosts.

V ONLINE ROGUE-AP DETECTION SYSTEM

We design a system for online detection of rogue APs. This system consists of three major components as illustrated in Fig. 5. The data capturing component collects incoming and outgoing packet headers. These packet headers are then passed on to the online detection engine, where WLAN hosts are detected using the algorithms described in the previous sections. Once a WLAN host is detected, its IP address is looked up from an authorization list for rogue-AP detection. We next describe the online detection engine, the core component in the system, in more detail. Afterwards, we describe how to identify ACK-pairs in real time and obtain inter-ACK time distributions a priori (required by the sequential hypothesis test with training).

V.I Online Detection Engine

The online detection engine makes a detection on a per host (or IP address) basis. Since TCP data packets and ACKs come in on a per flow basis and a host may have multiple simultaneous active TCP flows3, the online detection engine maintains a set of data structures in memory, each corresponding to an active TCP flow. We name the data structure as an unpacked-data-packet queue since it stores the information on all the data packets that have not been acknowledged by the receiver. Each item in a queue represents a data packet in the corresponding active flow. It records the sequence number (4 bytes), the timestamp (8 bytes) and size (2 bytes) of the packet. In addition, the online detection engine also records the latest ACK for each TCP flow in memory. These information is used to identify ACK-pairs as follows. For each incoming ACK, the online detection engine finds its corresponding unpacked-data-packet queue (using a hash function for quick lookup) and then matches it with the items in the queue to identify ACK-pairs. Once an ACK-pair is identified, depending on whether training data is available, it is fed into the sequential hypothesis test with or without training to determine whether the host uses WLAN. The memory requirement of the online detection system mainly comes from storing the unpacked-data-packet queues.

Each queue contains no more than M items, where M is the maximum TCP window size (since an item is removed from the queue once its corresponding ACK arrives). In our experiments, we find that most queues contain a very small number of items (see Section 7.3), indicating that the memory usage of this online detection system is low.

V.II Online Identification of TCP ACK-pairs

As described earlier, two successive ACKs form an ACK-pair if the inter-arrival time of their corresponding data packets at the monitoring point is less than a threshold T (chosen as 240 μs or 400 μs in our system, see Section 7). In addition to the above condition, we also take account of several practical issues when identifying ACK-pairs. First, we exclude all ACKs whose corresponding data packets have been retransmitted or reordered. We also exclude ACKs due to expiration of delayed-ACK timers if delayed ACK is implemented (inferred using techniques in [25]). This is because, if an ACK is triggered by a delayed-ACK timer, it is not released immediately after a data packet. Therefore, the inter-arrival time of this ACK and its previous ACK does not reflect the characteristics of the access link. Furthermore, to ensure that two ACKs are

successive, we require that the difference of their IPIDs to be no more than 1. We also restrict
that the ACKs are for relatively large data packets (of size at least 1000 bytes), to be consistent with the assumption of our analysis (in Section 3). Last, we require that the inter ACK time of an ACK-pair to be below 200ms.

VI DISCUSSIONS

We next discuss several issues related to rogue AP detection.

VI.I Locating Rogue APs

Our approach to detecting a rogue AP also helps to locate the rogue AP. Let us consider a common scenario in which a WLAN host is connected to a rogue AP, which is connected to an access router via one or multiple switches. In this scenario, the rogue AP can be located using the following steps. First, a network manager detects the IP address of the WLAN host at the monitoring point, and then locates the access router of this host based on the host's IP address and the subnet addressing structure. From the ARP table at the access router (which stores the mapping between an IP address and its corresponding MAC address), the network manager further determines the MAC address of the WLAN host. Afterwards, the network manager uses the identified MAC address to obtain its corresponding switch port by SNMP querying the first downstream switch connected to the access router (this is through the switch table at the switch, which stores the mapping between a MAC address and a switch port). Last, the network manager sequentially queries downstream switches (if any) to locate the switch port (and hence the physical location) of the rogue AP.

VI.II Rogues by Authorized Users

Our scheme can easily detect rogue APs installed by hosts not authorized to use WLAN. We next discuss the case that rouges are installed by hosts authorized to use WLAN. We consider two types of local networks: purely wireless networks (i.e., all IP addresses are allowed to use wireless connections) and mixed networks (i.e., networks supporting both Ethernet and wireless connections).

Purely wireless networks. In such a network, a wireless host A may set up another wireless card as a rogue AP for an illegitimate host B (as described in Section 7.4). In this case, packets from B will have the IP address of A, which is an authorized WLAN address. Therefore, our scheme does not detect this type of rogue directly. However, since host B connects to the Internet through two wireless hops, its traffic characteristics will differ from those through a single wireless hop and those through Ethernet, and hence can be detected through traffic analysis. An accurate detection scheme for this type of rogue is left as future work.

Mixed networks. In such a network, we consider two scenarios. In the first scenario, the IP address blocks for Ethernet and WLAN connections do not overlap. Then a host will have different IP addresses for its Ethernet and WLAN connections. In this scenario, if a host authorized to use both
Ethernet and WLAN installs a rogue AP on its Ethernet connection, the host obtains an IP address in the Ethernet block and the associated rogue AP will be easily detected by our scheme. In the second scenario, the IP address blocks for Ethernet and WLAN connections overlap. Then a host may maintain the same IP address for both Ethernet and WLAN connections. Similar to the first scenario, we can detect rogue APs that provide hosts Internet connection using two wireless hops through traffic analysis. However, a host authorized to use WLAN may also set up a rogue AP on its Ethernet connection for itself to connect to the Internet. This type of rogue cannot be detected by our scheme or traffic analysis (since this host only uses a single wireless hop).

VI.III Possible attacks to our approach

Our approach is based on inter-ACK times. It is effective for the common scenario where a rogue AP is installed by an innocent user (for convenience or flexibility). It is also robust against MAC-address spoofing attacks. However, a rogue AP may change the inter-ACK times to elude being detected by our algorithms. For instance, it may reduce the inter-ACK times by buffering ACKs first and then releasing them in a batch in order to disguise the traffic as Ethernet traffic. Such a camouflage, however, will inevitably increase local RTTs (i.e., the portion of RTT inside the WLAN). Therefore, we may combine inter-ACK time and local RTT measurement to detect such a camouflage. An effective scheme is left as future work.

CONCLUSIONS

In this paper, we have proposed two online algorithms to detect rogue access points, based on real time passive measurements collected at a gateway router. Both algorithms exploit the fundamental properties of the 802.11 CSMA/CA MAC protocol and the half duplex nature of wireless channels to differentiate Ethernet and WLAN TCP traffic. Central to both algorithms are sequential hypothesis tests that determine a host's connection type in real time by extending our earlier TCP ACK-pair techniques [25]. One algorithm requires training sets, while the other does not. Extensive experiments in various scenarios and over hosts with various operating systems have demonstrated the excellent performance of our approach: the algorithm that requires training provides rapid detection and is extremely accurate; the algorithm that does not require training detects 60%-76% of the wireless hosts without any false positives; both algorithms require computation and storage well within the capability of commodity equipment. Furthermore, our scheme can detect connection switching and wireless networks behind a NAT box. Last, our scheme remains effective for hosts with high CPU, hard disk or network utilizations.

REFERENCES

[1] Air Defense, Wireless LAN Security.http://airdefense.net.
[2] AirMagnet. http://www.airmagnet.com.[3] Airwave, Airwave Management Platform.http://airwave.com.
[4] Cisco Wireless LAN Solution Engine (WLSE). http://www.cisco.com/en/US/products/sw/cscowork /ps3915/.
[5] Host AP. http://hostap.epitest.fi.
[6] http://www.endace.com.
[7] Microsoft Windows 2000 TCP/IP implementation details, http://www.microsoft.com/technet/itsolutions/network/deploy/depov g/tcpip2k .mspx.
[8] NetStumbler. http://www.netstumbler.com.
[9] Rogue Access Point Detection: Automatically Detect and Manage Wireless Threats to Your Network. http://www.proxim.com.

[10] A. Adya, V. Bahl, R. Chandra, and L. Qiu. Architecture and techniques for diagnosing faults in ieee 802.11 infrastructure networks. In Proc. ACM MOBICOM, September 2004.
[11] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill. Enhancing the security of corporate Wi-Fi networks using DAIR. In Proc. ACM MOBISYS, 2006.
[12] V. Baiamonte, K. Papagiannaki, and G. Iannaccone. Detecting 802.11 wireless hosts from remote passive observations. In Proc. IFIP/TC6 Networking, Atlanta, GE, May 2007.
[13] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland. Rogue access point detection using temporal traffic characteristics. In Proc. IEEE GLOBECOM, Dec 2004.
[14] G. Casella and R. L. Berger. Statistical Inference. Duxbury Thomson Learning, 2002.
[15] L. Cheng and I. Marsic. Fuzzy reasoning for wireless awareness. International Journal of Wireless Information Networks, 8(1), 2001.
[16] S. Garg, M. Kappes, and A. S. Krishnakumar. On the effect of contention-window sizes in IEEE 802.11b networks. Technical Report ALR-2002-024, Avaya Labs Research, 2002.
[17] IEEE 802.11, 802.11a, 802.11b standards for wireless local area networks.
http://standards.ieee.org/getieee802/802.11.html.
[18] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In Proc. IEEE Symposium on Security and Privacy, May 2004.
[19] C. Mano, A. Blaich, Q. Liao, Y. Jiang, D. Salyers, D. Cieslak, and A. Striegel. RIPPS: Rogue identifying packet payload slicer detecting unauthorized wireless hosts through network traffic conditioning. ACM Transactions on Information Systems and Security, to appear.
[20] Packet trace analysis.
http://ipmon.sprintlabs.com/packstat/packetoverview.php.
[21] P. Sarolahti and A. Kuznetsov. Congestion control in Linux TCP. In Proc. USENIX02, June 2002.
[22] A. N. Shiryaev. Probability. Springer, 2nd edition.
[23] K. Thompson, G. Miller, and R. Wilder. Wide-areaInternet traffic patterns and characteristics. IEEE Network, 11(6):10–23, Nov./Dec. 1997.
[24] A. Wald. Sequential Analysis. J. Wiley & Sons, 1947.
[25] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley.Identifying 802.11 traffic from passive measurements using iteraltive Bayesian inference. In Proc. IEEE INFOCOM, 2006.
[26] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley. Classification of access network types: Ethernet, wireless LAN, ADSL, cable modem or dialup? In Proc. IEEE INFOCOM, March 2005.
[27] H. Yin, G. Chen, and J. Wang. Detecting Protected Layer-3 Rogue APs. In Proceedings of the Fourth IEEE International Conference on Broadband Communications, Networks, and Systems (BROADNETS), Raleigh, NC, September 2007.