Isolation of Multiple Anonymous Attackers in Mobile Networks

Brian Ricks and Patrick Tague

Carnegie Mellon University
{brian.ricks and patrick.tague}@sv.cmu.edu

Abstract. Many mobile wireless networks unintentionally provide opportunity for attackers to launch anonymous attacks or spoof other users, often without fear of being caught. It's often ideal for network carriers to block all traffic from an attacker, not just the attack traffic, for example to stop any concurrent attacks which cannot be detected by the carrier. We present an approach to detect common attacks at the access point, and leverage this with packet clustering to block all traffic originating from attackers during an attack. To achieve packet clustering, we utilize received signal strength at the access point to properly cluster attack packets according to each unique attacker, and further classify all other packets according to these clusters. Our approach is designed with attacker and legitimate user mobility in mind, low memory overhead, and is scalable to many simultaneous attackers. Our experimental results show very high classification accuracy, sensitivity and specificity.

1 Introduction

Preventing malicious behavior is an important challenge for network carriers. Such behavior can not only be detrimental to a carrier's legitimate customers, but can also be a liability issue for the carrier if such attacks are traced back to the carrier network as the source. While in traditional networks blocking such a malicious user may have entailed simply blocking the interface to which that user is connected, the problem becomes more difficult in modern wireless networks, where users are not physically connected to the network, but rather wirelessly connected to *access points* (APs). To further complicate the situation, wireless users are often mobile, such as with cellular networks, and they may hop between different access points in the carrier network.

Mobile networks which support and serve users using public shared key, or open networks, have additional challenges. In these networks, users are trusted simply if they know the shared key, or users are not trusted at all. Often times authentication reduces to mapping parameters to users, such as MAC address or IP address. As such mappings are easily spoofed by an attacker, for example using MAC addresses is basically an honor system type approach, it in essence reduces the network to one which comprises anonymous users. In other words, under a network environment of weak authentication or no authentication at all, we can treat the users as anonymous. Not only can this result in attackers launching attacks in which the carrier network does not know the source, but it also opens the door for attackers to spoof legitimate users during an attack. This culminates into the question: “I, as a mobile network carrier, can detect an ongoing attack, but I don’t know from whom the attack is originating. How can I completely block such an anonymous attacker from my network while they are being malicious? And can I do this without accidentally blocking legitimate users?”

We take an approach to answer these questions by first looking back at why some common attacks are difficult to detect accurately, and how next generation architectures may help us to defeat the attacks in ways that are not possible in traditional networks. Then we take a look at some additional properties of next generation mobile networks that may help us to block all traffic from attackers while they are launching attacks over such networks.

Our approach can be broken down into two steps. The first step is attack detection near the sources of attacks, as opposed to the target server or edge routers, exploiting the architecture common to next generation mobile networks. Our approach brings the detectors as close as possible to the end users: right at the first hop. We show how to detect with very high accuracy common flooding attacks which rely on source IP spoofing, such as TCP SYN flooding attacks, by using cross-layer packet header inspection, a feature of next generation mobile networks. As TCP SYN flooding attacks are well researched in the literature, we will use that attack class as a case study in this paper.

The second step is attacker isolation, based on clustering of the ongoing attack(s). For this step we utilize *Received Signal Strength Indicator (RSSI)* to classify all anonymous traffic through the AP as either belonging to an attacker or to a benign user. This is achieved without the carrier network ever knowing the identity of the attacker(s). Blocking all attacker traffic is beneficial to the carrier network as the attacker may be launching concurrent attacks, not all of which the network carrier may be able to detect. This approach works for multiple attackers launching concurrent attacks, and can provide information in terms of the total number of unique attackers launching attacks at any given time.

We validate our approach through rigorous experimentation, which gives very promising results, with very high classification accuracy, sensitivity and specificity both for attacker traffic and user traffic. We further show that our approach is easily deployable, with few parameters, all of which can be tweaked within a broad range without affecting classification accuracy significantly. This allows for quicker deployment without optimal parameter tuning.

The rest of the paper is organized as follows: Section 2 covers relevant background, including related work. Section 3 gives an overview of our approach, divided into detection and isolation steps. Section 4 provides a methodology of our experimental process, results, and in-depth analysis of the results. Section 5 presents some challenges and limitations of our current approach, and finally, Section 6 summarizes our work.

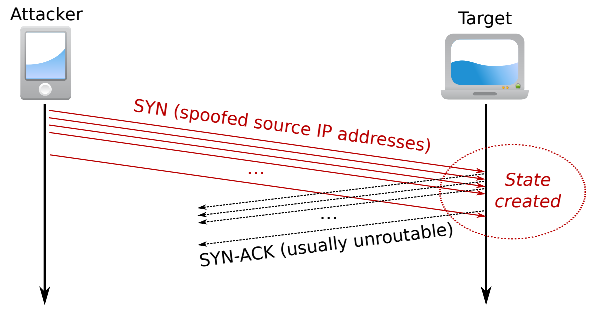
Attacker
Target
SYN (spoofed source IP addresses)
...
State created
...
SYN-ACK (usually unroutable)

Fig. 1. Illustration of a SYN Flooding attack. State is created each time a SYN packet with a unique source IP address is received at the target.

2 Preliminaries

Here we define important background and discuss relevant related work.

2.1 SYN Flooding Attacks

A *SYN Flooding* attack is class of TCP-based denial-of-service attacks, with the end goal of the attacker to disrupt service at the target TCP server. The attacker accomplishes this by exploiting the natural flow of the three-way TCP connection handshake. Under normal circumstances, a TCP connection is established using the following high-level description:

- The client sends a SYN packet to the TCP server. The TCP server, upon receiving this packet, creates state called a *half-open connection*.
- The TCP server sends back a SYN-ACK packet to the client. The TCP server uses the source IP address of the SYN packet as the destination IP address of the SYN-ACK packet.
- The client, upon receipt of the SYN-ACK packet, sends to the TCP server an ACK packet. This completes the connection, and the half-open connection state at the TCP server is deleted.

The attacker exploits the first step, by sending many SYN packets at once, with the goal of creating one half-open connection state at the TCP server for each SYN packet sent. To accomplish this, the attacker spoofs the source IP address of all generated SYN packets: a different, unique source IP address per SYN packet. If the attacker can send enough SYN packets, the half-open connection buffer will completely fill up before some timeout period, potentially denying service to legitimate clients trying to establish a TCP connection.

Detection of SYN flooding attacks traditionally has utilized statistical methods to model the flooding behavior. In Wang et al. [11] and Ling et al. [6], many of the same assumptions are utilized as our approach, such as cross-layer packet header inspection and detector placement near the source.[1] In Wang et al. [11], SYN flooding attacks are detected by correlating SYN/FIN packet pairs. Their approach however is reliant on the attacker not being aware of the method of detection, and the detector can be defeated by simply generating SYN/FIN pairs to defeat the statistical correlation. Their approach focuses specifically on TCP SYN flooding attacks, and not on attacker isolation.

In Ling et al. [6], ratio of SYN and SYN+ACK packets at an edge router are used to detect a possible SYN flooding attack coming from an intranet connected to the edge router. If an anomaly is detected, then source IP addresses of potentially malicious SYN packets are checked for reachability. While the approach has low computational overhead, it maintains state, and thus an attacker could attack the detection system not only by stateholding attacks, but by inducing the system to ping many potential end hosts for reachability, which could result in a detection system induced denial-of service.

Xiao et al. [13] assumes the detector at the destination TCP server. Their approach also assumes that half-open connections are either due to network congestion or a SYN flooding attack, and similar to Ling et al.[6] uses probing to detect potential SYN floods from suspicious half-open connections. This implies additional bandwidth overhead, though their approach tries to limit attacks on the detection system by sampling a subset of half-open connections as more half-open connections are added to the TCP server.

2.2 Received Signal Strength

Received Signal Strength Indicator, or *RSSI*, is a measurement, taken at a wireless receiver, of the perceived power of an incoming radio signal.[2] RSSI measurements are unitless but correspond to measurements in mW or dBm, and the higher the value, the stronger the received signal. In the real-world, accuracy of RSSI measurements can vary greatly from vendor to vendor [1, 7].

In this paper, we refer to *per-packet RSSI*, defined as the RSSI measurement taken during the preamble stage of the last 802.11 frame received which comprises a single IP packet. Note that additional noise present during measurement, for example from other transmitters, should not arbitrarily affect the resulting per-packet RSSI value, as any additional signal strong enough to do so should result in a collision with the incoming frame and resulting loss of that frame.

Previous work using RSSI as a metric mostly falls into the category of spatial localization [9, 12]. Our work departs in that we are using RSSI not to locate the attacker spatially, but to isolate the attacker's traffic from the network. Sheng et al. [8] uses RSSI measurements to detect MAC address spoofing, however their assumptions would not be suitable in an environment where attacker mobility is present, as any assumptions on attacker mobility could be easily defeated by an attacker by changing their mobility patterns.

[1] Here detector placement is at edge routers, which will be one hop away from possible intranets, but usually not one hop away from the users themselves.

[2] By "incoming radio signal", we mean the *strongest* incoming radio signal within the receiving band for a receiver.

Yang et al. [14] and Faria and Cheriton [2] also use RSSI measurements to detect attacks, but factor in mobility. Yang et al. [14] uses a binary partitioning scheme and thresholding to detect spoofing attacks, but requires a training phase, making real-time selective packet blocking difficult. Faria and Cheriton [2] uses multiple concurrent RSSI values from APs and applies sets of user-defined rules to the resulting tuples, called *signalprints*, to detect both spoofing and flooding attacks, though this does not extend to attacker isolation specifically.

3 Blocking Anonymous Attackers During an Attack

We discuss our approach for blocking anonymous attackers during an attack by breaking it down into two parts: detection of common attacks, and the isolation of potential attacker traffic. Figure 2 gives an illustration of our approach deployed on a generic mobile network.

3.1 Detecting Common Attacks at the Carrier Network

A *common attack* is a well-known attack whose origins are traditionally in the global internet. These attacks normally assume that the network architecture is that of the global internet, a static, wired network of computers. Indeed, the IP protocol suite which drives the internet was built upon these same assumptions.

One goal of common attack detection in next-generation mobile networks is to leverage additional information, and exploit physical constraints within these networks to build a new set of network assumptions which can then be applied to possibly detect such attacks. We currently build our network assumptions based on two characteristics present in next-generation mobile networks:

- Leveraging cross-layer opportunities afforded to us through next generation architectures that do not follow the strict separation of OSI layers [11, 6, 5]. This allows us to probe specific higher layers, such as the network and transport layers, from lower layers, such as the MAC or physical layer.
- Exploiting specific network topology present in these mobile networks. Users wirelessly connect via access points (APs) to the network, making the APs one hop away from the users. Placing detectors at the APs has three distinct advantages. First, the task of attack detection is distributed to multiple entities (APs) within the network as opposed to a centralized entity. Second, detector placement at the APs provides opportunity for reduced overhead in only detecting attacks feasible in their locality. Third, this placement puts the detectors under the jurisdiction of the carrier, as opposed to possibly many entities if placing them outside the carrier network at edge routers [11, 6]. This makes such a solution more viable to real-world deployment.

These characteristics can be applied to extract information from higher layer packet headers at lower layers, and from a specific point in the carrier network to classify a packet as benign or a potential attack packet (Figure 2).

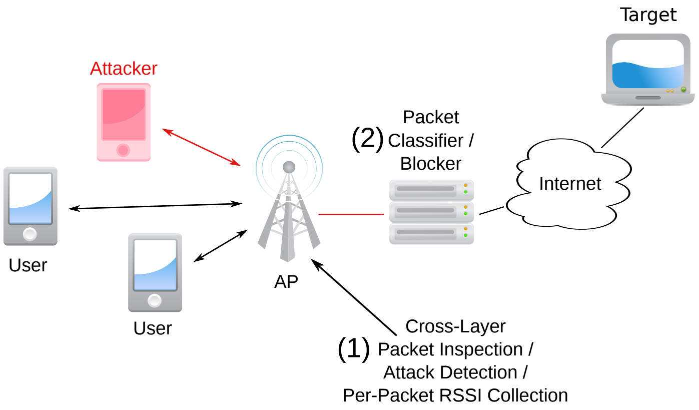
Target
Attacker
(2) Packet Classifier / Blocker
Internet
User
User
AP
(1) Cross-Layer Packet Inspection / Attack Detection / Per-Packet RSSI Collection

Fig. 2. Example mobile network with our approach deployed. The attacker and users are mobile, connected to the AP shown. The AP incorporates the attack detection step of the approach (**(1)** - Section 3.1), while the Packet Classifier handles packet classification and blocking (**(2)** - Section 3.2).

The advantages are two-fold: first, cross-layer packet header inspection allows lower layers a global view of an incoming packet. Second, higher layer packet header information may have specific context at a specific point in the carrier network, but this context may be lost by the time the packet reaches its destination. For example, if such context is in the transport layer and specific to nodes that are not the source or destination, then cross-layer header inspection must be leveraged to extract the contextual information.

To illustrate this approach to packet detection, we discuss a common class of attacks: those which utilize source IP spoofing.

Attacks Utilizing Source IP Address Spoofing This class of attacks usually falls into the category of denial-of-service attacks, and relies on the attacker spoofing the source IP address of packets they generate, to give the appearance of many packets from seemingly many users. The goal is to deny service to legitimate users by exhaustion of some resource, for example server resources or bandwidth. SYN flooding (Section 2.1) and ICMP flood attacks are two well-known examples of denial-of-service attacks which utilize source IP address spoofing.

Our approach works in the following way. Each time an AP receives a packet, a quick cross-layer check is applied to verify that the source IP address of the packet is assigned to a user currently connected to that AP. Traditionally APs are layer 2 devices, and thus do not check (or understand) layer 3 header information. In a cross-layer environment however, no such constraint exists, and thus the AP is trusted for layer 3 inspection. In this case, the detector only needs to understand the layer 3 (network) header. Note that in anonymous environments, there may be no reliable way to differentiate users, as IP addresses or MAC addresses can be spoofed [3]. We also do not make any assumptions that an authentication mechanism is in place.[3]

If the source IP address is determined to be assigned to a user currently connected to the AP (we call this a *valid* IP address), then the packet is deemed not malicious. However, if the source IP address is determined to not belong to any user currently assigned to the AP (an *invalid* IP address), then the packet is potentially malicious. However, the following are also possible:

- If the source IP address belongs to the carrier network's subnet, then there is a probability that the AP may not be aware that a legitimate user is connected to it. For example, a configuration error between APs may have resulted in a mobile user hopping to this AP from another, but without an exchange of state between the APs.
- If the source IP address is outside the carrier network's subnet, then there may be a configuration issue with the user.

To increase our confidence that a potentially malicious packet correlates to an attack, the detector temporarily saves state related to this packet. Then within a small window, if another packet is deemed potentially malicious, it is compared against the previous packet state that was saved. If the source IP addresses do not match, then we deem that an attack is underway. If the source IP addresses match, then the temporary packet state is deleted and new state saved for the current packet. This guarantees that the detector only stores state related to one packet at a time, preventing stateholding attacks against the detector [4].

3.2 Isolating Attacker Traffic

While detection of the attacker's malicious activities is good to block those activities specifically, this still won't prevent an attacker from performing other activities, and otherwise using the network during an attack. While it may be beneficial for a carrier to completely block an attacker during an attempted attack for many reasons, one important reason is that an attacker may be launching concurrent attacks, not all of which may be detectable by the carrier network.

In traditional wired networks, blocking an attacker completely from the first hop is trivial. Simply block the interface which the attacker is connected on.[4] However, in wireless networks, all users connected to an AP share the same physical medium, and thus we must utilize other information to correlate an attacker's malicious packets with other traffic originating from the attacker. The anonymous environment and potential mobility of users further complicates this. The carrier network does not know where any of the users physically are located outside the APs that they are connected to.

[3] Any authentication mechanism deployed can be used to provide further information to our approach, but an authentication mechanism may also be vulnerable to defeat, and such a discussion is outside the scope of this paper.

[4] Furthermore, when the attacker changes its logical identity, it appears as a new interface, so the L2/L3 linkage is broken.

To solve these problems, we employ an approach that leverages layer 1 information provided at the AP. More specifically, each time an AP receives a packet, providing that it is determined that an attack is underway (Section 3.1), the per-packet RSSI value is recorded. This value is then used to classify the packet as originating from an attacker or other. If the origination is from an attacker, then the packet is dropped.

The Clustering Procedure Our clustering procedure forms clusters based on attack packet RSSI value similarity. The clusters themselves represent a single unique attacker's most recently transmitted attack packet; thus each cluster only has a single data point at any given time, keeping the memory footprint low. We don't keep older data points because of the temporal dependency between an attacker's movement and transmission of packets: older packets simply do not reflect the current mobility state of an attacker.

In an ideal environment, one in which all users are stationary and there is no drift in RSSI measurements, clusters would only comprise members whose RSSI values are identical. However in real-world mobile networks, we have to factor in both attacker mobility and RSSI measurement drift [1, 7].

Similarity of two attack packets require a *similarity metric*, to quantify the similarity, and a *similarity threshold*, to provide binary classification (does an attack belong to a cluster or not?). We formally define the similarity metric, s_{ij}, as

$$s_{ij} = |PKT_i^{RSSI} - CLS_j^{RSSI}|, \tag{1}$$

where PKT_i^{RSSI} is the per-packet RSSI of attack packet i we are classifying as measured by the AP, and CLS_j^{RSSI} is the most recent per-packet RSSI measurement, taken by the AP, that was assigned to cluster j.

We formally define our similarity threshold, e_{ij}, as

$$e_{ij} = \Delta t_{ij} * d_u + d_{ap}, \tag{2}$$

where Δt_{ij} is the time difference between the timestamp of a new attack packet i recorded at the AP and the timestamp of cluster j, d_u is a drift constant which models RSSI drift due to user mobility, and d_{ap} is a drift constant related to RSSI sampling precision at the AP. For stationary users, e_{ij} reduces to a constant term $e_{ij} = d_{ap}$. An attack packet is assigned to the first cluster j in which $s_{ij} <= e_{ij}$. If an attack packet does not meet this criteria for any cluster, a new cluster is created, and the packet assigned to it.

Non-attacker packets are classified using the same approach, except we do not actually assign them to a cluster. Only attack packets (packets detected from the steps in Section 3.1 for example) are assigned to clusters after classification.

When attack packets are first detected for a new attack, we do not consider these packets as comprising an attack until a certain number of *sequential* packets have been assigned to the same cluster (the packets themselves are still dropped as they are invalid). Two attack packets are considered sequential if they are both assigned to the same cluster. We consider an attack to have started when

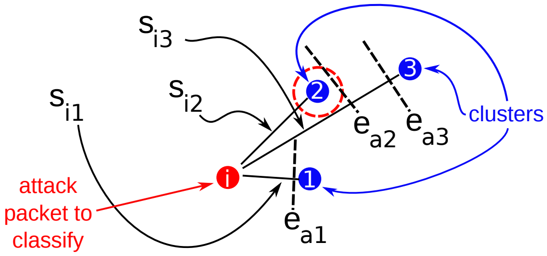

Fig. 3. Illustration of the clustering procedure. An attack packet is assigned to the first cluster j in which $s_{ij} <= e_{ij}$. The goal is not to find the 'closest' cluster, but whether the attack packet belongs to an existing cluster or a new one. In this figure, the solid line segments represent s_{ij}, with shorter line segments representing greater similarity. Dashed line segments represent e_{ij}, with greater distance from attack packet i representing a larger threshold. Visually, $s_{ij} > e_{ij}$ if the dashed line segment for cluster j resides between the cluster and attack packet i. Here, attack packet i would be assigned to cluster 2. The key is that similarity alone does not determine cluster assignment, but the relationship between packet RSSI similarity and user mobility.

the number of sequential packets observed, p_{num}, reaches a threshold parameter, c_{pre}. More formally, an attack is considered ongoing when $p_{num} > c_{pre}$

Clusters are not permanent: a threshold parameter, c_{to}, is used to determine when an attack is no longer ongoing (ceases to exist). If no new attack packets have been assigned to a cluster within c_{to}, then the cluster is deleted. More formally, if an attack packet is assigned to a cluster at time t, then another attack packet must be assigned to the same cluster at some future time t_{next}, such that $t_{next} < (t + c_{to})$.

Note that our clustering procedure is independent of the detection approach. As long as there is some method to differentiate attack packets from all other packets, this clustering procedure (and the traffic isolation approach in general) can be used.

4 Experiments

Here we present our experimental methodology and results from our experiments.

4.1 Methodology

We implemented our approach in the OMNeT++ simulation framework [10]. We modeled a WiFi network with a single AP, and variable attackers and other mobile users of the network. While WiFi networks are not next-generation in themselves, all the basic building blocks for our required architecture are included in OMNeT++'s Inet WiFi simulation models. We extended these models with various support which we needed, such as cross-layer packet inspection.

We ran a total of five experiments on the simulated mobile network, each one increasing the number of attackers and benign users. The first scenario served as

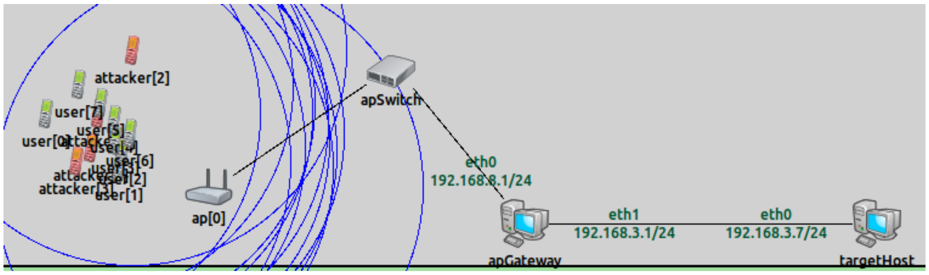

Fig. 4. Scenario 5 at $t = 0$. Notice that all the attackers and users are grouped tightly together. They all move southbound at differing velocities. The attackers are shown in red.

a baseline, with only a single attacker. The other 4 scenarios consisted of between 1-4 attackers, and 5-8 benign users. Each scenario was executed for 120 seconds. The parameters were set as follows: $d_u = 0.1$, $d_{ap} = 0$, $c_{pre} = 2$, $c_{to} = 1s$.

The attackers each launched SYN flooding attacks, with a start time of anywhere between 1-4 seconds. This was done intentionally to illustrate the progression of individual attacker detection (Figure 8). Both the attackers and benign users send a steady stream of UDP traffic.

Both the attackers and benign users are mobile, with velocities randomly selected from a uniform distribution of the range 0.2-2.5 m/s. This roughly corresponds to a range between a slow walk to a brisk jog. All attackers and benign users remain at their originally selected velocities throughout a scenario.

The following list explains the metrics used for interpreting the results:

- **Accuracy:** The percentage of correct classifications of all traffic.
- **Sensitivity:** The percentage of correct classifications of traffic originating from attackers. The complement of this is the false negative rate.
- **Specificity:** The percentage of correct classifications of traffic originating from benign users. The complement of this is the false positive rate.

4.2 Experimental Results

Table 1 gives an overview of the classification tests performed against our approach. Some highlights include a 100% accuracy rate when only an attacker is present. This provides a good baseline to make sure d_u is set high enough to properly put all attack packets into the correct clusters. In the case of a single attacker, there should only be one cluster. Even as we add more attackers, and more benign users, the accuracy, sensitivity, and specificity remain very high.

Figure 5 illustrates the e_{ij} and s_{ij} plots for the scenario 1 TCP SYN packets from the attacker. s_{ij} correlates to the path the attacker is following, in this case a straight line. We set d_u high enough as to compensate for the attacker's mobility, as can be seen by the relation of the s_{ij} and e_{ij} plots in Figure 5. Setting d_u (or d_{ap}) too high can result in a higher false positive rate.

Scenario	Description	Total Packets	**Accuracy**	**Sensitivity**	**Specificity**
1	*1 Attacker* *0 Users*	1186	100%	100%	N/A
2	*1 Attacker* *5 Users*	9080	99.912%	100%	99.899%
3	*2 Attackers* *5 Users*	11258	97.735%	94.059%	99.406%
4	*3 Attackers* *5 Users*	12360	98.115%	95.887%	99.513%
5	*4 Attackers* *8 Users*	18580	99.128%	99.215%	99.086%

Table 1. Results of 5 scenarios varying the number of legitimate users and attackers within a basic WiFi network.

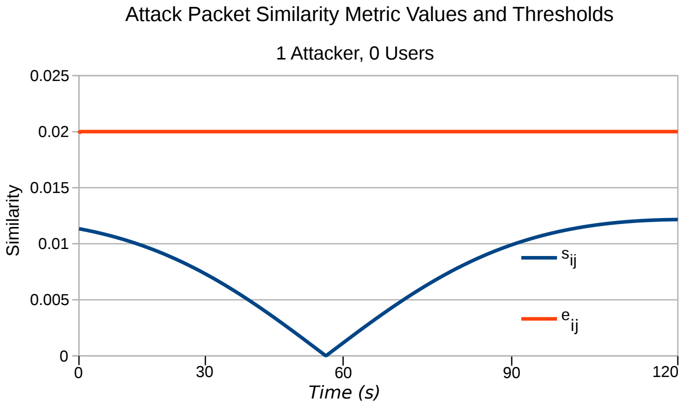

Fig. 5. e_{ij} (orange - top) and s_{ij} (blue - bottom) plots for scenario 1.

False positives can occur when an attacker is transmitting an attack packet at nearly the same time as a benign user transmitting a packet, and the per-packet RSSI readings of both packets are very close, resulting in a low s_{ij}. As can be seen from in Figure 6, this does not occur very often, though the probability of this occurring does increase slightly as the number of benign users increases. Adding another point of reference, such as another AP nearby, can dramatically lower the false positive rate.

False negatives can occur when d_u is set too low. For example, if we set $d_u = 0.2$ and rerun scenario 3, our sensitivity increases from 94.059% to 98.294%. The reason is that d_u cannot always fully compensate for all attacker mobility, such as if an attacker is moving faster than anticipated, or sending attack traffic much slower than what is expected for the attack type. The trade-off is often a lower specificity, in this case decreasing from 99.406% to 98.798%.

Figure 8 shows the number of attacker clusters versus time, for scenario 5. The 4 attackers start the SYN flooding attack at slightly offset start times from

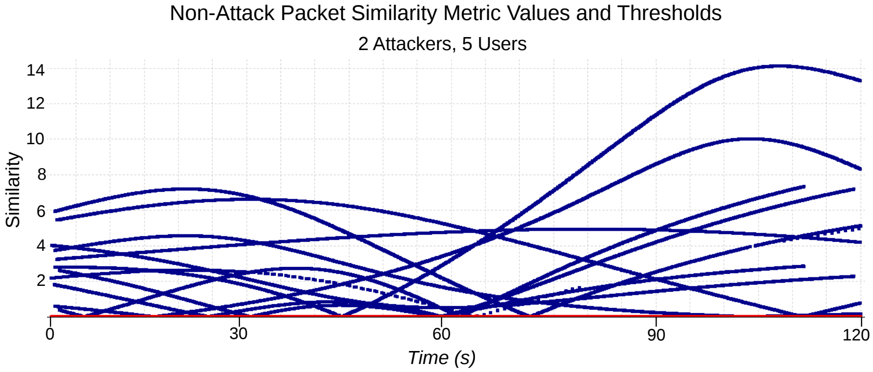

Fig. 6. e_{ij} (red - bottom) and s_{ij} (blue) versus time during scenario 3 for benign user packets. Notice that because the s_{ij} plots are above the similarity thresholds, these packets will not be classified as originating from an attacker.

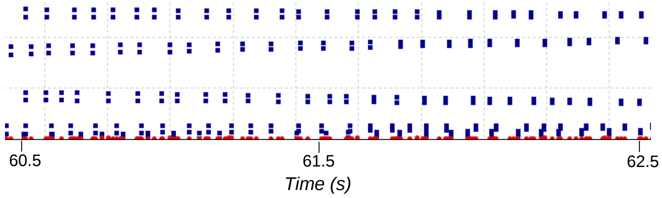

Fig. 7. e_{ij} (red - bottom - circles) and s_{ij} (blue - squares) of a 2 second zoomed in section of Figure 6

one another, resulting in the steep slope at the beginning of the graph. There is a one-to-one correspondence between attackers and clusters. In cases where d_u is set artificially low (or d_{ap} in some cases), many more clusters than attackers may be created at specific times, resulting in a very large decrease in accuracy.

5 Limitations and Future Work

Our current approach is limited to blocking attacker traffic during an ongoing attack. We are currently working on extending this approach to predict malicious user traffic after an attack has ceased. This will allow for the approach to continuing blocking multiple concurrent attacks which may actually be ongoing after the detectable attacks have ceased.

It is possible that a carrier network's subnet will be so large that an attacker could successfully launch an attack only using source IP addresses that are within this subnet. However such a technique is easily defeatable by the carrier network, by simply giving APs access to a lookup table with all currently allocated IP addresses and to which AP these users are currently connected to. Any mismatch would indicate an invalid source IP address with a particular AP.

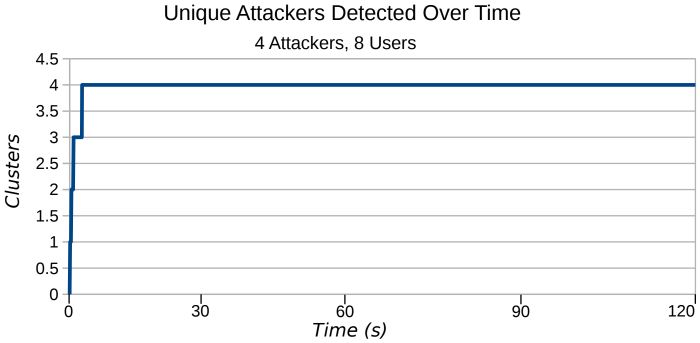

Fig. 8. Graph of the number of clusters versus time for scenario 5. The number of clusters corresponds to the number of unique attackers. Notice how the calculated number of clusters remains at 4 once all 4 attackers are detected.

The accuracy of attacker traffic isolation is dependent on the physical medium in which the wireless signals propagate, and also on the precision of the measurements themselves. While d_{ap} introduces robustness to the approach, imprecise measurements will require this parameter to be increased, which comes at a cost of a potentially higher false positive rate.

Another limitation is the assumption that attackers will not selectively adjust transmit power during an attack to separate their attack traffic from their benign traffic, as seen from the AP. Various solutions are being actively explored, such as using additional passive sensors to record RSSI measurements to supplement the measurements from the AP. Related to this is the real-world environment in general. We believe our approach is robust given APs which record RSSI accurately and precisely, and adding passive sensors for RSSI measurement should provide additional robustness against the non-ideal medium of the real-world. Experimentation on our physical network testbed is planned as future work.

6 Conclusion

We introduced an approach to isolate mobile attacker traffic during attacks originating from next generation mobile networks. This approach works under the assumption that all users of the network are either anonymous, such as networks relying on shared public key, or can defeat any authentication scheme deployed on the network to spoof other benign users. Our approach uses a combination of detecting common attacks at the access points, and clustering of attack traffic using RSSI to form clusters corresponding to unique attackers. Performing packet classification over these clusters resulted in a vast majority of attacker-originated traffic being successfully blocked, with very little legitimate user traffic blocked. Our approach is scalable up to many mobile attackers and users.

References

1. Chen, Y., Terzis, A.: On the mechanisms and effects of calibrating rssi measurements for 802.15.4 radios. In: Silva, J., Krishnamachari, B., Boavida, F. (eds.) Wireless Sensor Networks, Lecture Notes in Computer Science, vol. 5970, pp. 256–271. Springer Berlin Heidelberg (2010)
2. Faria, D.B., Cheriton, D.R.: Detecting identity-based attacks in wireless networks using signalprints. In: Proceedings of the 5th ACM Workshop on Wireless Security. pp. 43–52. WiSe '06, ACM, New York, NY, USA (2006)
3. Guo, F., Chiueh, T.: Sequence number-based mac address spoof detection. In: Proceedings of the 8th International Conference on Recent Advances in Intrusion Detection. pp. 309–329. RAID'05, Springer-Verlag, Berlin, Heidelberg (2006)
4. Handley, M., Paxson, V., Kreibich, C.: Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In: Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10. pp. 9–9. SSYM-01, USENIX Association, Berkeley, CA, USA (2001), http://dl.acm.org/citation.cfm?id=1267612.1267621
5. Iannucci, B., Tague, P., Mengshoel, O.J., Lohn, J.: Crossmobile: A cross-layer architecture for next-generation wireless systems. Tech. Rep. CMU-SV-14-001, Carnegie Institute of Technology (Mar 2014)
6. Ling, Y., Gu, Y., Wei, G.: Detect syn flooding attack in edge routers. International Journal of Security and its Applications 3(1) (Jan 2009)
7. Lui, G., Gallagher, T., Li, B., Dempster, A., Rizos, C.: Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization. In: Localization and GNSS (ICL-GNSS), 2011 International Conference on. pp. 53–57 (June 2011)
8. Sheng, Y., Tan, K., Chen, G., Kotz, D., Campbell, A.: Detecting 802.11 mac layer spoofing using received signal strength. In: INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. pp. – (April 2008)
9. Sugano, M.: Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. In: Wireless and Optical Communications. pp. 1–6. IASTED/ACTA Press (2006)
10. Varga, A., Hornig, R.: An overview of the omnet++ simulation environment. In: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. pp. 60:1–60:10. Simutools '08, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2008)
11. Wang, H., Zhang, D., Shin, K.: Detecting syn flooding attacks. In: Proceedings of IEEE INFOCOM - Volume 3. pp. 1530–1539 (2002)
12. Wessels, A., Wang, X., Laur, R., Lang, W.: Dynamic indoor localization using multilateration with rssi in wireless sensor networks for transport logistics. Procedia Engineering 5(0), 220 – 223 (2010), eurosensor XXIV Conference
13. Xiao, B., Chen, W., He, Y., Sha, E.H.M.: An active detecting method against syn flooding attack. In: The 11th IEEE International Conference on Parallel and Distributed Systems (ICPADS-05). vol. 1, pp. 709–715 (Jul 2005)
14. Yang, J., Chen, Y., Trappe, W.: Detecting spoofing attacks in mobile wireless environments. In: Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on. pp. 1–9 (June 2009)