On the Detection of Signaling DoS Attacks on 3G Wireless Networks

Patrick P. C. Lee, Tian Bu, and Thomas Woo

Abstract— **Third Generation (3G) wireless networks based on the CDMA2000 and UMTS standards are now increasingly being deployed throughout the world. Because of their complex signaling and relatively limited bandwidth, these 3G networks are generally more vulnerable than their wireline counterparts, thus making them fertile ground for new attacks. In this paper, we identify and study a novel Denial of Service (DoS) attack, called *signaling attack*, that exploits the unique vulnerabilities of the signaling/control plane in 3G wireless networks. Using simulations driven by real traces, we are able to demonstrate the impact of a signaling attack. Specifically, we show how a well-timed low-volume signaling attack can potentially overload the control plane and detrimentally affect the key elements in a 3G wireless infrastructure. The low-volume nature of the signaling attack allows it to avoid detection by existing intrusion detection algorithms, which are often signature or volume-based. As a counter-measure, we present and evaluate an online early detection algorithm based on the statistical CUSUM method. Through the use of extensive trace-driven simulations, we demonstrate that the algorithm is robust and can identify an attack in its inception, before significant damage is done.**

I. Introduction

The targets of most denial-of-service (DoS) attacks so far are wireline end-points, whose prevalence provides vast opportunities for an attacker to explore and launch new attacks. As the roll-out of nation-wide wireless data networks continues, we expect more types of DoS attacks will start targeting wireless networks. Currently, third generation (3G) wide-area wireless networks based on the CDMA2000 [24] and UMTS [25] standards are widely deployed. As of December 2005, there were over 300 million CDMA subscribers worldwide[6]. Emerging 3G data standards, such as EV-DO and HSDPA, promise to deliver broadband mobile Internet services with peak rates of 2.4 Mbps and 14.4 Mbps, respectively. The number of data subscribers is projected to reach a billion before 2010 [6]. As the number of data-capable wireless endpoints escalates, they will become susceptible targets of new DoS attacks in the near future.

Apart from the sheer number of mobile endpoints, a multitude of other factors also contribute to making a 3G wireless network more vulnerable to attacks. These include:

- *Limited wireless link bandwidth:* As opposed to most wireline links, 3G wireless links tend to have much lower capacity thus it takes significantly less traffic to overload the link.
- *High signaling overhead:* With existing 3G standards, to transfer a similar amount of data, a lot more signaling messages/handshakes is needed for a wireless network than that in a wireline network. For instance, in order to improve the utilization of limited radio resources, a radio channel is only allocated to a mobile when there is data to transfer, and it will be revoked after an inactivity timeout. Such dynamic channel allocation and revocation procedures introduce lots of signaling operations.
- *Heavy control processing:* The hierarchical nature of current 3G (CDMA2000 or UMTS) networks places certain critical system functions such as power control, resource allocation, paging, etc. on a few infrastructure elements. The Radio Network Controller (RNC) and the base stations (BS) are involved in these activities for each mobile. By necessity, the engineering of these network elements is typically based on a certain load profile that is derived from the projected traffic patterns and behaviors of mobiles. Any operational deviation from these design assumptions can cause significant overload condition, and potentially non-graceful degradation.

In a nutshell, 3G wireless networks are significantly more fragile than wireline networks. To begin, most of the wireline DoS attacks would still apply to a wireless network. In addition, the above unique vulnerabilities of 3G networks can be exploited by new forms of wireless-specific DoS attacks.

In this paper, we introduce a novel DoS attack termed the *signaling attack*, which seeks to overload the control plane of a 3G wireless network using *low-rate, low-volume* attack traffic, based on some of the aforementioned 3G-specific vulnerabilities. Unlike conventional DoS attacks that focus on the data plane, the signaling attack creates havoc in the signaling plane of a 3G network by repeatedly triggering radio channel allocations and revocations. To accomplish this, an attacker first sends a low-volume packet burst to a mobile. If the mobile does not currently have a radio channel, the network will allocate a new one to complete the data transfer. After an inactivity timeout, the radio channel is torn down to recycle it back for others' use and help preserve the mobile energy that will otherwise be wasted on maintaining the channel. Immediately after the channel release, the attacker sends another low-volume packet burst to the mobile so as to trigger another radio channel establishment. By repeatedly doing so at appropriately timed periods, this can generate a considerable number of signaling operations. As detailed in Section II, each channel establishment/release requires the RNC and BS to process more than 20 signaling messages. Launching this against large number of mobiles can easily introduce an excessive amount of signaling messages. The potential damage includes (1) overloading of RNC and BS,

P. Lee is with the Dept of Computer Science, Columbia University (email: pclee@cs.columbia.edu). T. Bu and T. Woo are with Bell Labs, Alcatel-Lucent, (emails: {tbu, woo}@research.bell-labs.com).

leading to reduced system performance, (2) denial of service to legitimate signaling messages due to congestion in the signaling paths, and (3) shortening of the mobile battery life.

As opposed to current DoS attacks that generate aggressive traffic, the signaling attack can be achieved with low-rate, low-volume traffic. Thus, the signaling attack can effectively evade detection by today's intrusion detection/prevention systems, which are effective mostly against flooding-based DoS attacks.

To understand the damage caused by the signaling attack, suppose that a 3G wireless network has inactivity timeout set to 5 s[1] and that an attacker generates a 40-byte packet burst. By sending packet bursts periodically at a time slightly larger than 5 s, the attacker generates only 64bps attack traffic, which is invisible to volume-based detection systems. If the attacker is using a cable modem with 1Mbps uplink bandwidth, then it can simultaneously attack approximately 160K mobiles, a number potentially sufficient to bring down a wireless network infrastructure that serves a large metropolitan area such as New York City. Note that this signaling attack can also be mounted to other emerging wide-area networks such as 802.16/WiMAX that share the same vulnerability (see [17]).

We propose an online early detection mechanism for signaling attack by formulating it as a change-point detection problem, which aims to identify the sources of signaling attacks by monitoring any abnormal behavior against profiled benign behavior. Our detection algorithm is based on the *non-parametric CUSUM test*, which does not require the parameterization of the network traffic distribution of interest and has been shown to be asymptotically optimal for change-point problems [4], [5], [20], [27]. To make the algorithm robust against intelligent attackers that attempt to increase the detection delay and hence aggravate the damage, our detection mechanism is derived in a way that the detection delay depends only on the amount of additional signaling load due to the attack but not on anything else, such as the attack strategy.

To summarize, this paper makes the following contributions. We identify a novel signaling DoS attack specific for 3G wireless networks and demonstrate its severity through simulation driven by synthetic and real traces. The attack is unique in the sense that it overloads the control plane with only low-rate, low-volume attack traffic, and hence makes conventional signature or volume-based detection schemes ineffective. To combat this, we present a novel online early detection algorithm that can identify the attackers before they cause any major damage. Through extensive trace-driven evaluation, we demonstrate that our detection algorithm has a short detection delay while rarely generating false alarms.

The remainder of the paper proceeds as follows. Section II presents the vulnerabilities of the signaling plane in a general 3G wireless network and explains how they are susceptible the signaling attack. Based on trace-driven simulation, Section III demonstrates the severe damage resulting from the signaling attack. In Section IV, we present a CUSUM-based detection mechanism to identify the source of a signaling attack. In Section V, we evaluate the effectiveness of our detection mechanism. Section VI addresses possible enhancements to our current work and the reaction mechanism. Section VII reviews related work on different forms of DoS attacks in wireline and wireless networks, and we conclude in Section VIII.

[1]It is the default value in many deployed networks.

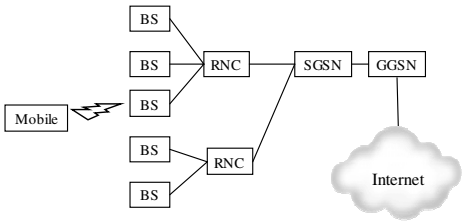

Fig. 1. UMTS Network Architecture.

II. Signaling attacks on 3G Wireless Networks

In this section, we first overview the network elements of a 3G network architecture and their inter-connections. Here, we use the UMTS network architecture as our example. We focus on the signaling procedures among the network elements for radio channel establishment and release. We then demonstrate how an attacker may exploit these signaling procedures to overload the control plane.

Note that the signaling attack can be launched to CDMA2000 and 802.16/WiMAX networks as well. In the interest of space, we refer readers to [17] for the discussion.

A. UMTS Network Architecture

Figure 1 shows the typical architecture of a UMTS wireless network. We first describe two of its main components: the Gateway GPRS Support Node (GGSN) and the Serving GPRS Support Node (SGSN). The GGSN is a GPRS network entity that serves as the mobile wireless gateway between an SGSN and the Internet. When a mobile successfully authenticates and registers with the network, a Point-to-Point (PPP) link is set up between the GGSN and the mobile. On the other hand, the SGSN is responsible for sending data to and from mobile stations, in addition to maintaining information about the location of a mobile and performing authentication for the mobile. Typically, there are multiple SGSNs, each of which serves the GPRS users physically located in its serving area.

Another key component of a UMTS network is the Radio Network Controller (RNC), which is the point where wireless link layer protocols terminate. The RNC provides the interface between a mobile communicating through a Base Station (BS) and the network edge. This includes management of radio transceivers in BS equipment (radio resource control), admission control, channel allocation, as well as management tasks such as handoffs between BSs and deciding power control parameters. The functionalities of a BS include wireless link transmission/reception, modulation/demodulation, physical channel coding, error handling, and power control. In this hierarchical architecture, multiple mobiles communicate with a BS, and multiple BSs communicate with an RNC, and multiple RNCs talk to the SGSN/GGSN.

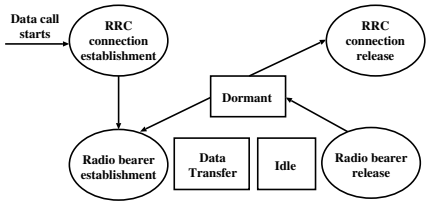

Fig. 2. UMTS data call. Multiple RABs may be established within one RRC connection.

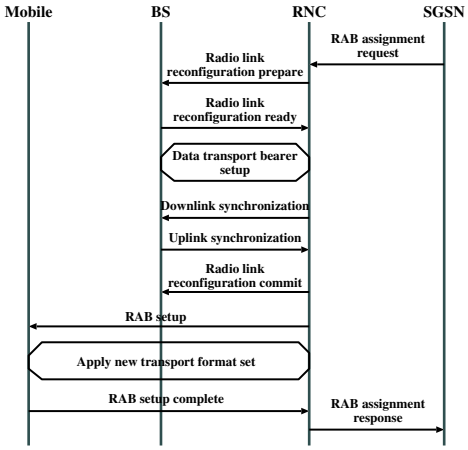

Fig. 3. UMTS Radio Bearer Establishment

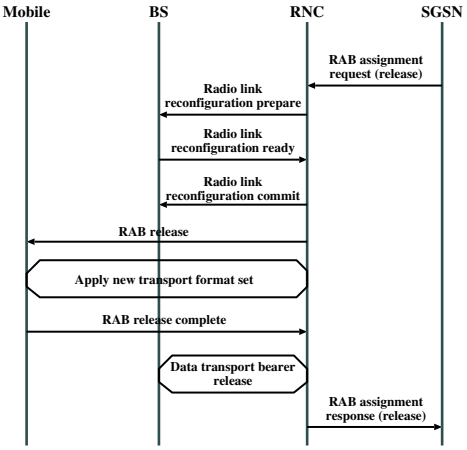

Fig. 4. UMTS Radio Bearer Release

B. Signaling for Radio Resource Control

The data service session provided by the network to mobile is often referred to as a *data call* that starts from the mobile connects to the network for data service and stops when the mobile disconnects from the network. Figure 2 shows the steps of a UMTS data call. The paging stage is not presented as it is necessary only when the network originates a data call. We also skip the authentication and security events that are not relevant in the context. Detailed specification of a UMTS data call can be found in [1].

A Radio Resource Control (RRC) connection is first created. While there is only one RRC connection during a data call, the network may establish one or more Radio Access Bearers (RAB) within the single RRC connection in an on-demand fashion. RABs are the actual radio resources for data communication. They are released after a timeout period for inactivity so that they can be reused by other mobiles. In addition, the mobiles may extend their battery life by releasing the idle RAB because maintaining a RAB requires periodic channel condition updates that consume much energy.

When new data arrives for a mobile without a RAB, an RAB establishment procedure is invoked to allocate radio resources for data delivery. The procedure varies depending on the type of RAB being established or released. Figure 3 shows the procedure for establishing a synchronized RAB. There are about a total of 15 signaling messages being processed by the RNC. The processing load would be even higher in order to support fast handoff.

The allocated RAB is released after an inactivity timeout. Similarly, the release procedure will vary depending on the type of the RAB. Figure 4 shows the procedure for the release of a synchronized RAB. There are a total of 12 signaling messages being processed by the RNC. Similarly, the support of fast handoff would result in even more signaling steps.

C. Signaling Attacks on UMTS Networks

Due to the signaling overhead required for RAB setup/release, an attacker may seek to trigger excessive amount of signaling messages in order to overload an RNC and potentially the BSs. This can be done by regularly sending low-volume bursts at appropriately timed periods such that immediately after a RAB is torn down due to inactivity, a burst arriving from the attacker will trigger a new RAB establishment.

The attack may cause severe damage to the control plane: (a) the signaling path between the RNC and the BS is congested with the setup/release messages associated with each RAB; (b) the RNC processor is tied up with maintaining states and processing signaling messages and is thus overloaded. The consequence of (a) is that valid signaling messages may not receive allocation of resources, causing it to be dropped by the RNC due to insufficient buffers and/or excessive delay that leads to timeout. The consequence of (b) is that overloading an RNC can effectively deny legitimate services to the mobiles being served by the RNC. Another side effect of the signaling attack is the potential draining of the mobile battery. Normally, for power conservation, a mobile switches to a low-power idle or dormant state when there is no packet to be sent or received. Since low-volume bursts are sent regularly, mobiles would stay active longer than necessary.

As shown in Section I, if the RAB inactivity timeout is 5 s, then by generating 40-byte packets to 160K mobiles periodically at a time slightly larger than 5 s, or equivalently 64 kbps attack traffic only, all RNCs serving a metropolitan area such as New York city can be brought down. The low-volume nature of the signaling attack makes it hard to be detected by conventional intrusion-detection mechanisms that are designed to defend against flooding-based DoS attacks.

III. Impact of the Signaling Attack

To demonstrate the impact of the signaling attack, the ideal approach is to attack an operational 3G wireless network and observe performance degradation or even network breakdown. However, the experiments on a real network are not feasible due to both economical and legal reasons. We may also collect data traces and corresponding signaling traces from 3G operational networks and simulate the signaling attack by inserting hypothetical attack traffic into the traces. However, data traces from live 3G networks are not currently available due to sensitivities around privacy and competition. Therefore,

we consider two types of traces for the purpose of demonstration: (1) synthetic traces derived from the 3G traffic models presented in the literature and (2) real traces collected from a campus-wide wireless network.

A. Simulation with Synthetic Traces

Based on the 3G traffic models described in [2], [9], [18], we construct a discrete-event traffic simulator that generates synthetic traces for a UMTS network. The trace generation is composed of three levels. In the *session level*, the simulator creates a new user session according to a Poisson arrival process with mean 1 s, such that each session is of type voice or data with probabilities 0.4 and 0.6, respectively. We assume that a voice session has an exponentially distributed duration with mean 120 s, while a data session lasts until all session data has been transferred, where the data volume follows a lognormal distribution with parameters μ=11.1170 s and σ=1.3818 s. In the *burst level*, each (voice or data) session consists of alternating ON/OFF periods such that packets are generated during the ON period only. Both ON and OFF durations are exponentially distributed, with the same mean 3 s for voice sessions and with means 0.2075 s and 12 s, respectively, for data sessions. In the *packet level*, packets are generated based on the distributions of inter-arrival time and packet size. Since the packet-level time variations are so small that they are insignificant to the inactivity timer of a radio channel, we assume that the packet-level parameters are constant. Thus, we fix the packet inter-arrival time and packet size to be 0.02 s and 31 bytes, respectively, for voice sessions (based on the AMR 12.2 mode) and 8.3 ms and 480 bytes, respectively, for data sessions. Finally, we assume that each mobile is associated with a single user session, and that after the session is completed, the mobile remains connected to the wireless network for 30 minutes based on the observation in [15].

After generating the synthetic traces, we add an attacker that periodically injects additional low-volume packet bursts at every *attack interval*, which is set to be slightly larger than the inactivity timeout of a radio channel so as to maximize the chance of introducing signaling events. Specifically, the attacker randomly selects a subset of mobiles that are connected to the network. It then sends a 40-byte packet (e.g., a TCP packet with zero payload) to each of the attacked mobiles. Any attacked mobile that has been idle over the channel inactivity timeout will invoke a signaling event at the RNC for establishing a new radio channel. Here, we focus on a UMTS network that has inactivity timeout set to 5 s. It should be noted that this inactivity timeout is a tunable parameter that is set up by network operators once and rarely changes.

Figure 5 illustrates, over a 4-hour-long synthetic trace, the signaling load recorded every 15 minutes when different numbers of mobiles are attacked at each attack interval. It clearly shows the severe damage resulting from the signaling attack. For example, when 80 mobiles are attacked at each attack interval, the signaling load increases by 2.5 times as compared to the no-attack case.

Also, the 4-hour-long synthetic trace contains 2.4 GB of normal traffic, while the additional traffic for attacking 80 mobiles accounts for (14400 / 5 $\times$ 80 $\times$ 40 bytes) / 2.4GB = 0.4% of the total traffic only. Among the attack traffic, about 90% of which triggers a signaling event at an RNC. This shows that an RNC can be heavily overloaded with low-volume traffic.

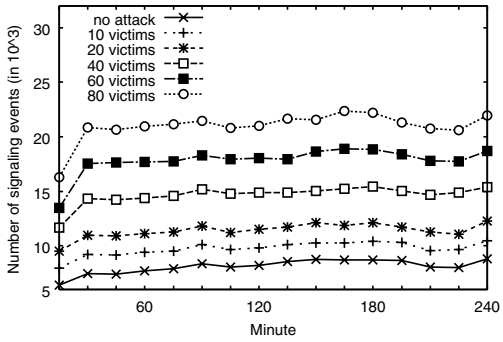

Fig. 5. Signaling load under a signaling attack using a synthetic trace.

B. Simulation with Real Traces

As noted in Section III-A, the packet-level time scales are much smaller than the inactivity timer of a radio channel. Therefore, although the wireless access medium may disturb the packet transmission delay, the channel inactivity and hence the RAB setup/release events are less influenced by link delay variance of the access medium. Instead, they are mainly influenced by the packet arrival pattern resulting from user and application behaviors as well as network transport layer protocols. Thus, regardless of the access medium, the impact of a signaling attack that we observe should be similar to that in the previous synthetic-trace simulation.

Due to the unavailability of real traces from a 3G operational network, we instead take the traces collected from the IEEE 802.11b wireless local area network (LAN) at Dartmouth College in Fall 2003 [15] for our evaluation purpose. We emphasize that by no means do we suggest that these traces are ideal for a 3G wireless network. In order to use the traces to drive the simulation of a 3G network, we first assume that all hosts associated with the access points are mobiles in a 3G network. We then simulate the signaling load at an RNC before and after the signaling attack.

We use the same setup as in Section III-A for injecting attack traffic, i.e., at each attack interval that is slightly larger than 5 s, an attacker sends a 40-byte packet to a random set of mobiles that are connected to a wireless network.

Figure 6 illustrates the signaling load when different numbers of mobiles are attacked at each attack interval, using the traces collected on November 3, 2003. As in Section III-A, the signaling load is significantly increased due to the signaling attack. For instance, by periodically attacking 80 mobiles, the signaling load of the RNC increases more than five times, while (not shown in the figure) only less than 0.6% additional traffic is introduced. We also repeat the simulation using traces collected from different days and the results are similar.

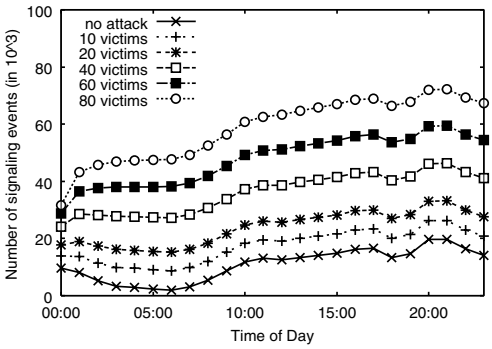

Fig. 6. Signaling load under a signaling attack using the real traces collected from Dartmouth College.

C. Discussion

When a malicious host launches signaling attacks, it might not know the exact set of IP addresses that have been assigned to *active mobiles* (i.e., the mobiles being connected to the wireless network). However, the IP address segments of major wireless service providers are public information [11]. Recent studies [7], [21] also discuss how to identify active mobiles in a cellular data network. To simplify our simulation without compromising our key results, we only insert attack traffic for active mobiles.

Since the bandwidth required for attacking a single mobile is very small, the attacker may choose to send packets to all or a large portion of the IP addresses of a wireless network. Any packet targeting an inactive mobile is dropped by GGSNs (UMTS) or PDSNs (CDMA2000), whereas any packet reaching an active mobile that has its radio channel released would trigger a signaling event for a channel setup. In this case, the signaling attack can still be achieved with low-volume attack traffic even the active mobiles only occupy a small portion of the IP address segment. For example, in the real-trace simulation in Section III-B, if we assume that the active mobiles only occupy 10% of the IP address segment, the attacker only needs 6% additional traffic to increase the signaling load of an RNC by five times.

IV. Detecting Signaling Attacks

In order to defend against signaling attacks, we propose to monitor all data traffic that enters and leaves a 3G wireless network for detecting any ongoing signaling attacks and their originators. Our *online* detection algorithm is based on the statistical *cumulative sum (CUSUM)* test, whose objective is to identify any anomaly that deviates from normal behavior as early as possible.

In this section, we first describe what data samples should be used as the input to the detection algorithm and where they should be collected. We then overview the CUSUM test and demonstrate how to apply the CUSUM test for robust detection of signaling attack with the appropriate choices of parameters.

A. Data Sample Collection

We now explain how data samples are collected for the purpose of attack detection. We define a *remote host* to be a node that originates downlink traffic to a mobile within a wireless network. Note that a remote host can be either a node in the Internet or another mobile in the same wireless network. We uniquely identify a *flow* (a.k.a. *session*) by the address pair of a mobile and a remote host. We refer a packet to be *inbound* (resp. *outbound*) if it is transmitted from (resp. to) a remote host to (resp. from) a mobile. We treat a packet as inbound if the remote host is a mobile itself. We say a remote host *triggers* a *virtual setup* if it initiates a packet to a flow that has not transmitted any inbound or outbound packet over the last inactivity timeout.

It is important to note that virtual setups are measured at the flow level. They do not necessarily lead to any actual radio channel setup (and hence any signaling event) since a mobile does not release its radio channel if there have been data transfers between the mobile and other remote hosts in the last inactivity timeout. However, the virtual setup provides a measure of the intention of a remote host to launch a signaling attack. If the remote host triggers a significant number of virtual setups over a short period of time, then there is a strong evidence that the remote host attempts to mount a signaling attack, regardless of how many signaling operations are actually caused by these virtual setups.

In our analysis, we only focus on inbound attacks, i.e., the signaling attack is always originated from remote hosts. Although a mobile may frequently establish and release radio channels with other non-mobiles, the signaling events are associated with a single mobile only. Hence, the increase in the RNC load is very limited.

For a given remote host, we define the *inter-setup time* as the time interval between two adjacent virtual setups (which can be on the same flow or on two different flows) that are triggered by the remote host. The smaller the inter-setup time is, the higher likelihood that the corresponding remote host is mounting a signaling attack. Therefore, for each remote host, we monitor a sequence of its inter-setup time samples $\{t_1, t_2, \cdots\}$ in the CUSUM test (see Section IV-C). We then use the abrupt decrease of the inter-setup time as an indicator of a signaling attack.

B. Placement of Detection Points

In order to collect input data for the detection algorithm, we have to monitor all IP packets entering and leaving a wireless network, decode their source and destination addresses, and compute the inter-setup times. Although we may collocate the detection elements with an RNC/SGSN for monitoring all mobiles served by the RNC/SGSN, neither the RNC nor the SGSN works at the IP layer, leading to extra efforts for extracting IP packets from either radio frames (in the RNC) or encapsulated packets (in the SGSN). An ideal place for data collection would be between the GGSN and the Internet where all traffic is composed of IP packets. However, the inter-mobile communication always turns around at the GGSN and is thus not available for data collection in this setting. To tackle this problem, we suggest two solutions. The first one is to add a packet sniffer that passively captures all packets entering and leaving the GGSN and pass them to the detection unit.

Another solution is to configure the GGSN so that all inter-mobile traffic is routed to the detection unit and then back to the GGSN. This loop-back feature is supported by the GGSN and being used for firewalling inter-mobile traffic today.

Some security measures such as firewalls and intrusion detection systems may have been placed between the GGSN and the Internet for protecting a wireless network from other general attacks. We may then implement our detection algorithm as an additional detection module in an existing detection system and take advantage of the traffic policies that have already been built into the system for reacting to the detected attacks.

C. Overview of the CUSUM Test

We first overview the CUSUM test and motivate how it is well-suited for intrusion detection. Here, we consider its *non-parametric version* that does not assume any a priori distribution of the data samples being considered. Thus, as long as the data samples are not extremely dependent, we are guaranteed the *asymptotic optimality*, i.e., the detection time is minimized among all possible detection schemes subject to a fixed worst-case expected false alarm rate [5], [27]. Detailed discussion of the CUSUM test and its non-parametric version can be found in [4], [5], [20], [27].

In the context of the signaling attack detection, for each remote host, the CUSUM test monitors a set of n inter-setup time samples $\{t_1, t_2, \cdots, t_n\}$[2]. Each inter-setup time sample t_n is assigned a score $z(t_n)$. When a sample t_n is available, we update the *CUSUM statistic* q_n as follows:

$$\begin{aligned} q_n &= \max(q_{n-1} + z(t_n), 0), \\ &\text{Take action if } q_n \geq h, \end{aligned} \tag{1}$$

where $h > 0$ is the pre-specified *CUSUM threshold*. Note that if an inter-setup time sample t follows a malicious behavior, then the expected score $E(z(t))$ should be positive so that q_n will eventually rise above threshold h. On the other hand, $E(z(t))$ should be negative when the data samples follow a benign behavior. We justify the choice of $z(t)$ in Section IV-D.

The CUSUM test is adequate for identifying any abrupt change of a benign behavior to a malicious behavior. To understand this, note that if a host is benign, $z(t_n)$ is negative (in the expected sense) and the corresponding q_n will stay around the zero value, regardless of how long the benign behavior has been observed. However, when the benign behavior turns to a malicious one, q_n increases and eventually surpasses the threshold. Therefore, the CUSUM test prevents an attacker from suppressing q_n with a long history of benign behavior. This ensures that the CUSUM test detects a malicious behavior in a timely manner.

D. Choice of Score $z(t)$

It is important to prevent an attacker from tricking the CUSUM test with an intelligent strategy. Thus, we seek to define $z(t)$ such that no attacker can evade detection as it causes extra signaling overhead, i.e., if any two attackers introduces the same expected rate of signaling attacks, then the expected delay to detect both attackers should be the same, regardless of their attack strategies. In other words, if any two attackers introduce the same number of virtual setups over time T, then they should have the same cumulative score $\sum z(t)$, which captures the behavior of all inter-setup time samples, when the CUSUM test is carried out at time T.

With this objective, we can formally prove that $z(t)$ must be a *linear function of* t. The detailed proof is shown in [17].

The formula for $z(t)$ can now be derived as follows. We consider a tunable parameter α, defined as the cut-off point such that if inter-setup time $t > \alpha$, it is likely to be a benign sample, while if $t < \alpha$, it is likely to be a malicious sample. Since $z(t)$ is linear, it attains maximum when $t = 0$. By setting $z(0) = 1$, we have $z(t) \leq 1$ for all t, and hence the ceiling of the CUSUM threshold $\lceil h \rceil$ denotes the minimum number of inter-setup time samples required to decide if a remote host is malicious. As a result, we can write $z(t)$ as

$$z(t) = \frac{-t}{\alpha} + 1. \tag{2}$$

E. Choice of h

For a given choice of $z(t)$ with $E(z(t)) < 0$, the probability that the CUSUM statistic q_n surpasses a positive value h is small but not zero either. However, the probability decreases as the value of h increases. Therefore, the false positive ratio is a decreasing function of h. On the other hand, q_n takes longer to surpass a large h even when $E(z(t)) > 0$ and hence the detection delay increases. The choice of h is to trade off between the false positive rate and the detection delay. Note that for some specific distributions (e.g., exponential) of data samples, h is often derived from the *average running length (ARL)* between two false positives that measures the false alarm tolerance. In the non-parametric setting (i.e., no a priori distribution of data samples is known), we will show how the empirical value of h can be obtained from the real traces in Section III-B.

F. Detection Algorithm

We now present the detection algorithm that monitors if any remote host is mounting a signaling attack, as shown in Algorithm 1. The detection algorithm is applied only to the remote hosts that are not marked as malicious attackers. Upon the arrival of a packet P (either inbound or outbound), the

Algorithm 1 Detection algorithm

```
for each arrival of packet P belonging to F = (M, R) do
  if R has not been marked "malicious" then
    if P is inbound & virtual setup triggered then
      if the setup is NOT the first one triggered by R then
        Set t = now - R's last setup time
        Set R.q = max{R.q + (−t/α + 1), 0}
        if R.q > h then
          Mark the remote host as "malicious"
      Set R's last setup time = now
```

[2]In [17], we validate via trace-driven simulation that the inter-setup time samples are not strongly dependent, so the condition of applying the CUSUM test is satisfied.

detection algorithm first determines the corresponding flow F, which identifies mobile M and remote host R. An inbound packet from R only triggers a virtual setup for flow F (Line 3) when no packet has been observed in flow F over last inactivity timeout. If the virtual setup is not the first one due to packets from R (Line 4), a new inter-setup time is computed (Line 5) and the CUSUM values associated with R are updated (Line 6) using Equations 1 and 2. R is marked malicious when the associated CUSUM statistic crosses the pre-selected threshold h (Lines 7 and 8). The most recent time that R triggers a virtual setup is tracked for computing the next inter-setup time (Line 9). Note that the initial values of the CUSUM parameters for all remote hosts are always set to zero.

V. Performance Evaluation

In this section, we evaluate our detection mechanism against the signaling attack based on trace-driven simulation. From Section III, we observe similar damage brought by the signaling attack using both synthetic and real traces. Thus, our following evaluation focuses on the real traces collected from Dartmouth College. While the wireless LAN traces are not ideal, our evaluation methodology can be directly applied when real traces from a 3G operational network is available.

Here, we consider a UMTS system whose inactivity timeout for a RAB is set to be 5 s.

A. Metrics

In evaluating an intrusion-detection system, there are three fundamental metrics that we can consider: (1) *false positive ratio*, the fraction of benign remote hosts that are mistakenly identified as malicious over all observed remote hosts, (2) *false negative ratio*, the fraction of malicious remote hosts that are missed over all observed remote hosts, and (3) *detection time*, the delay that the detection algorithm needs to identify a malicious remote host since it starts the signaling attack.

Here, we do not consider the false negative ratio. Because of the linearity of $z(t)$, the CUSUM decision depends only on the additional signaling load due to the signaling attack, regardless of what the attack strategy is. If there exists a false negative, then it implies that the signaling load generated by this missed attacker is within an acceptable level. In this case, the malicious hosts that are missed by our detection mechanism actually have little impact on the total signaling load. As a result, our following analysis focuses on the false positive ratio and the detection time.

The above metrics are quantified based on how we classify a remote host as benign or malicious. Here, we assume that all remote hosts in the original traces are benign, and that the remote hosts that we inject to the traces for generating attack traffic are malicious.

The false positive ratio and the detection time are intrinsically conflicting metrics that depend on how we choose the tunable parameters α and h for our detection algorithm (see Section IV). Intuitively, by choosing a small value of α and a large value of h, we reduce the false positive ratio, but in the meantime lengthen the detection time. We investigate the trade-off in the following subsections.

B. False Positive Ratio vs. Different Tunable Parameters

In this subsection, we analyze the false positive ratio of our detection mechanism with different values of α and h.

For a given a 24-hour trace, we set α to be the p-th percentile of inter-setup times of all remote hosts. In other words, we bound the proportion of inter-setup time samples that have positive $z(t)$ to be no more than $p\%$. Here, we consider the cases where p = 1, 5, 10, 15, and 20. We then compute the false positive ratio versus different values of h using our detection mechanism from the simulation.

Figure 7 illustrates the false positive ratio versus different values of h, together with the percentile values of the observed inter-setup time samples. In general, most remote hosts only trigger a few virtual setups with large inter-setup times. Thus, the false positive ratio in most cases is very small. For instance, when α is no higher than the 10th-percentile, the false positive ratio is less than 0.06% for $h \geq 3$.

C. Detection Time vs. Different Tunable Parameters

To assess the detection time, we add to the traces a malicious remote host that generates low-rate, low-volume packets to a number of active mobiles every attack interval (see Section III). We assume that the packets are randomly spaced over the attack interval. Also, we randomly choose the active mobiles to be attacked. We repeat the experiment 10 times with different seeds, and the measured detection time is averaged over the experimental instances.

Similar to the setting in Section III, we fix the attack interval to be slightly larger than the inactivity timeout (which is 5 s according to our assumption). We assume that the attacker targets 5 mobiles at each attack interval.

Figure 8 illustrates the detection time versus different values of h, where α is set at different percentiles of inter-setup time samples. For smaller values of α and larger values of h, the detection mechanism requires longer detection time to identify the attacker, but in the meantime it produces fewer false positives (see Figure 7). This suggests the trade-off between different combinations of α and h.

D. Evaluation for the Signaling Attack Defense

We now evaluate our detection mechanism for defending against the signaling attack. Since the false positive ratio is low according to our previous evaluation, we fix $\alpha = 7.1138$ s, which is the 10th-percentile inter-setup time, and $h = 3$.

Our setup is similar to that in Section V-C. In addition to the average detection time taken from 10 experimental instances, we also obtain the minimum and maximum detection times among those instances, and these bounds are shown as the endpoints of the vertical lines in the plots.

In Section III, we demonstrate how a single remote host can overload a RNC via the signaling attack. Figure 9 shows the detection time versus different numbers of victims. In most cases, the single attacker can be identified in less than 10 s since the signaling attack starts. As the attacker becomes more aggressive by attacking more mobiles, we need less detection time to identify this attacker. For instance, as shown

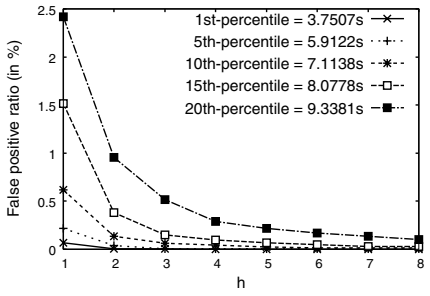

Fig. 7. False positive ratio vs. h.

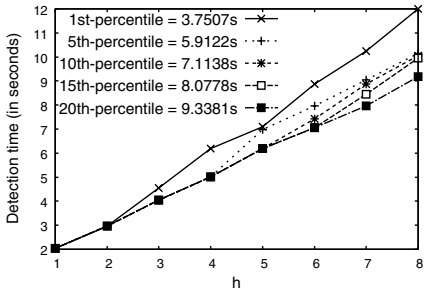

Fig. 8. Detection time vs. h.

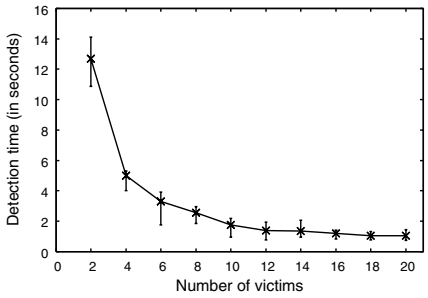

Fig. 9. Detection time vs. no. of attacked mobiles

in Section III, the attacker can double the signaling load of an RNC by periodically attacking 20 mobiles. However, such a signaling attack can be detected in less than one second. This shows our detection mechanism can quickly identify an attacker before it causes significant damage.

Furthermore, our detection mechanism remains robust toward different attack strategies that have the same attack intensity. The details can be found in [17].

VI. Enhancements to the Detection Mechanism and the Reaction

In this section, we address some enhancements to our detection mechanism so as to improve its scalability and enable it to handle more challenging attacks. We also describe a graceful approach for reacting to attacks that minimizes impact of false alarm yet maintain load at sustainable level.

A. Scaling up the Detection

Our detection mechanism keeps track of every remote host, and thus the storage requirement is proportional to the number of remote hosts, which can be up to the total number of hosts in the Internet. Although the amount of memory consumed by our simulation is not a vital concern due to the relatively small number of remote hosts in our traces as compared to the upper bound, our detection mechanism should be able to survive the abrupt surge of the number of remote hosts without exhausting the memory storage.

One approach is to bound the storage requirement. Suppose that we hash all remote hosts into a table of fixed size n. We define a *super node* that corresponds to the subset of remote hosts being hashed to the same entry. We then apply the detection algorithm to each super node rather than individual remote hosts. If a super node is not identified as malicious, obviously none of the remote hosts in the super node are malicious. On the other hand, if the super node is flagged malicious, we cannot tell if any of its associated remote hosts is malicious, because it is possible that some other remote hosts are malicious, or all remote hosts are benign but their aggregate behavior appears to be malicious.

In order to reduce false positives due to hash collisions, we introduce m parallel hash tables with independent hash functions (as in [8]). Each remote host is hashed to one entry in each of the m tables in parallel, i.e., any remote host is a member of m super nodes, each of which corresponds to a unique hash table (function). Now, a remote host is considered to be an attacker if its corresponding m super nodes are all marked malicious. In general, because the fraction of malicious remote hosts is small, a super node is less likely to have a malicious remote host. Moreover, the probability that a remote host collides with the same set of other hosts in all m hash tables decreases exponentially as m increases. Thus, the probability that a benign host has all its super nodes flagged malicious becomes very small. As a result, we can decrease (exponentially) the false positive ratio by increasing m.

By using m hash tables of size n, we can bound the total memory size to be $O(mn)$, while maintaining a small false positive ratio.

B. Attacks using Spoofed Addresses

Although attacks from spoofed address have been shown to be unpopular [28], attackers may still send malicious traffic using spoofed addresses. In this case, they not only hide their real origins, but also trick the detection mechanism to make wrong decisions. For instance, if an attacker always spoofs the same address for all attack traffic, the detection mechanism will identify the spoofed address as malicious and then block traffic coming from the spoofed address. This is only a problem when the spoofed address belongs to an active remote host that has legitimate communication with the mobiles.

There have been studies (e.g., [13]) on identifying and filtering packets with spoofed addresses that we may employ before applying the detection algorithm. In future work, we analyze how to extend our detection mechanism to address spoofing attacks as well.

C. Reaction Mechanism

In response to an identified malicious remote host, one possibility is to filter the subsequent traffic originated from that remote host. On the other hand, we may choose not to take any action since we can endure the presence of a signaling attack as long as the signaling load is within a sustainable level. This reduces the chance of blocking legitimate hosts that are falsely marked malicious (see Section V-B).

As the signaling load rises above a threshold, we may start blocking identified attackers that generate the most signaling load until the signaling load decreases to below the threshold. By doing so, we only take necessary actions to protect the

control plane from being overloaded while minimizing the chance to penalize legitimate remote hosts.

VII. Related Work

In this section, we review the related work on defending DoS attacks in both wireline and wireless networks.

Traditional DoS attacks are flooding-based such that an attacker generates *high-rate, high-volume* data traffic so as to deplete network resources using overwhelming data traffic. A number of defense mechanisms have been proposed for wireline networks, such as the pushback and traceback mechanisms [12], [22], [23]. Another class of DoS attacks is based on the *low-rate, high-volume* TCP attack [16], in which an attacker periodically generates high-volume packet bursts in order to force all TCP flows to repeatedly enter the retransmission timeout state. In this paper, we describe a *low-rate, low-volume* signaling attack that targets 3G or equivalent wireless infrastructures. Unlike the low-rate TCP attack, the signaling attack does not necessarily have any periodic pattern.

The flooding-based DoS attack is a common threat to both wireline and wireless networks. Other forms of DoS attacks that specifically target wireless networks include packet-forwarding disruption [3], [10], base-station impersonation [19], control-channel congestion via a sufficient number of SMS messages [7], and depletion of mobile batteries [21]. In particular, the DoS attack in [7] saturates the control channels for SMS communication, while that in [21] keeps a mobile in a high-battery-consumption state. Both of the attacks, similar to ours, can be achieved with low-volume attack traffic. On the other hand, the signaling attack considered in this paper exploits the heavy signaling overhead in 3G wireless networks.

Statistical online detection schemes have been studied by [14], [27] for countering DoS attacks. Specifically, [14] focuses on detecting malicious connection attempts based on Wald's test [26]. However, this detection scheme requires a priori probability distributions for the benign and malicious behaviors. In contrast, [27] propose a non-parametric CUSUM test to detect flooding-based DoS attacks based on periodic sampling. In this paper, we propose a different CUSUM-based method that is suitable for detecting the low-rate signaling attack and ensure that no attacker can intelligently escape from our detection mechanism.

VIII. Conclusions

We have presented a new DoS attack, called *signaling attack*, which targets 3G wireless networks. This attack works by exploiting the heavy-weight nature of signaling in 3G wireless networks. We have shown via trace-driven simulation that the signaling attack can substantially overload a wireless infrastructure using only minimal traffic. Because of its low-rate, low-volume property, the signaling attack can evade the detection of traditional counter-DoS systems. In view of this, we have proposed a statistical CUSUM-based detection mechanism to defend against the signaling attack. Using real-world traces, we have shown that our detection mechanism can identify the source of a signaling attack in a timely manner before the damage becomes aggravated, while producing very few false positives. In addition, our detection mechanism is robust as it depends solely on the additional signaling load and based on any assumed attack strategy.

References

[1] 3GPP. UTRAN Functions, Examples on Signaling Procedures (Release 1999), Jun 2002. TR 25.931 v3.7.0.

[2] 3GPP. IP Transport in UTRAN, Dec 2003. TR 25.933 v5.4.0.

[3] I. Aad, J.-P. Hubaux, and E. W. Knightly. Denial of Service Resilience in Ad Hoc Networks. In *Proc. of ACM MOBICOM*, Sep 2004.

[4] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[5] B. E. Brodsky and B. S. Darkhovsky. *Non-parametric methods in Change-Point Problems*. Kluwer Academic Publishers, 1993.

[6] CDMA Development Group. http://www.cdg.org.

[7] W. Enck, P. Traynor, P. McDaniel, and T. L. Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proc. of ACM CCS*, 2005.

[8] C. Estan and G. Varghese. New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice. *ACM Trans. on Computer Systems*, 21(3):270–313, Aug 2003.

[9] A.-B. Garcia, E. Garcia, M. Alvarez-Campana, J. Berrocal, and E. Vazquez. A Simulation Tool for Dimensioning and Performance Evaluation of the UMTS Terrestrial Radio Access Network. *Lecture Notes in Computer Science*, 2515, Nov 2002.

[10] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proc. of ACM MOBICOM*, Sep 2002.

[11] IANA. IP Address Services. http://www.iana.org/ipaddress/ip-addresses.htm.

[12] J. Ioannidis and S. M. Bellovin. Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *Proc. of the Network and Distributed System Security (NDSS)*, Feb 2002.

[13] C. Jin, H. Wang, and K. Shin. Hop-Count Filtering: An Effective Defense Against Spoofed DoS Traffic. In *ACM International Conference on Computer and Communications Security (CCS)*, Oct 2003.

[14] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *IEEE Symposium on Security and Privacy 2004*, May 2004.

[15] D. Kotz and K. Essien. Analysis of a Campus-Wide Network. In *Proc. of ACM MOBICOM*, Sep 2002.

[16] A. Kuzmanovic and E. Knightly. Low-Rate TCP-Targeted Denial of Service Attacks and Counter Strategies. *IEEE/ACM Trans. on Networking*, 14(5), Oct 2006.

[17] P. P. C. Lee, T. Bu, and T. Woo. On the Detection of Signaling DoS Attacks on 3G Wireless Networks. Technical Report, Bell Labs, Aug 2006.

[18] C. Lindemann, M. Lohmann, and A. Thümmler. Adaptive Performance Management for Universal Mobile Telecommunications System Networks. *Computer Networks*, 38(4), Mar 2002.

[19] U. Meyer and S. Wetzel. A Man-in-the-middle Attack on UMTS. In *Proc. of the ACM Workshop on Wireless Security (WiSe)*, Oct 2004.

[20] E. S. Page. Continuous Inspection Schemes. *Biometrika*, 41(1/2):100–115, Jun 1954.

[21] R. Racic, D. Ma, and H. Chen. Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery. In *Proc. of SecureComm*, Aug 2006.

[22] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Network Support for IP Traceback. *IEEE/ACM Trans. on Networking*, 9(3), Jun 2001.

[23] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. Single-Packet IP Traceback. *IEEE/ACM Trans. on Networking*, 10(6), Dec 2002.

[24] TIA/EIA/cdma2000. *Mobile Station - Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular Systems*. Washington: Telecommunication Industry Association, 1999.

[25] UMTS. *Release 5*. 3G Partnership Project.

[26] A. Wald. *Sequential Analysis*. New York, Dover Publications, 1973.

[27] H. Wang, D. Zhang, and K. G. Shin. Change-Point Monitoring for Detection of DoS Attacks. *IEEE Trans. on Dependable and Secure Computing*, 1(4), Dec 2004.

[28] V. Yegneswaran, P. Barford, and J. Ullrich. Internet Intrusions: Global Characteristics and Prevalence. In *ACM SIGMETRICS*, Jun 2003.