Can You Infect Me Now? Malware Propagation in Mobile Phone Networks

Chris Fleizach[†], Michael Liljenstam[‡], Per Johansson[†],
Geoffrey M. Voelker[†], and András Méhes[‡]

[†]University of California, San Diego
9500 Gilman Dr.
La Jolla, CA, USA 92093
{cfleizac,voelker}@cs.ucsd.edu,
pjohansson@ucsd.edu

[‡]Ericsson Research
Torshamnsgatan 23
SE-164 80, Stockholm, Sweden
michael.liljenstam@ericsson.com,
andras.mehes@ericsson.com

Abstract

In this paper we evaluate the effects of malware propagating using communication services in mobile phone networks. Although self-propagating malware is well understood in the Internet, mobile phone networks have very different characteristics in terms of topologies, services, provisioning and capacity, devices, and communication patterns. To investigate malware in this new environment, we have developed an event-driver simulator that captures the characteristics and constraints of mobile phone networks. In particular, the simulator models realistic topologies and provisioned capacities of the network infrastructure, as well as the contact graphs determined by cell phone address books. We evaluate the speed and severity of random contact worms in mobile phone networks, characterize the denial-of-service effects such worms could have on the network, investigate approaches to accelerate malware propagation, and discuss the implications of defending networks against such attacks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; D.4.6 [**Operating Systems**]: Security and Protection—*Invasive software*; I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete event*

General Terms

Security, Experimentation, Measurement

Keywords

Cellular phone networks, Worms, Epidemiology, Simulation, Measurement, Defenses

WORM'07, November 2, 2007, Alexandria, Virginia, USA.

1. INTRODUCTION

Mobile phones are the next frontier for malware. The combination of vulnerable platforms [20, 28], unsuspecting users [8], and explosive growth in potential victims [1] will inevitably attract propagating malware [5, 11]. As with the Internet, motivations will likely range from simple vandalism to identity and information theft, mobile phone spam, denial-of-service attacks, and potentially mobile bots, or *mobots*. The potential effects of virulent malware propagation on consumers and mobile phone providers are severe, including excessive charges to customers [6], deterioration of mobile phone services, public relations disasters, and ultimately loss of revenue for mobile phone providers.

In this paper, we evaluate the effects of malware propagating using communication services in mobile phone networks. We focus on malware designed to propagate quickly using communication services, since this situation represents a worst-case scenario for both consumers and network providers due to the resulting widespread infections and denial-of-service situations that might occur. Our goals are to model malware propagation in these networks under realistic scenarios to characterize its speed and severity, to understand how network provisioning impacts propagation and how propagation impacts the network, and to highlight the implications for network-based defenses against such malware.

To explore the range of possibilities of malware propagation on mobile phone networks, we have developed an extensive event-driver simulation environment that captures the characteristics and constraints of propagation in this environment. Since modeling the network is critical to understanding malware propagation behavior, we have developed a network topology generator that creates realistic topologies and provisioned capacities of the network infrastructure. Mobile phone networks offer a range of communication services, each with different propagation characteristics and network support. We model two prototypical services, a Voice over IP (VoIP) service in which malware can self-propagate by exploiting a vulnerability in the service implementation on the phone, and a Multimedia Messaging Service (MMS) in which propagation depends on user interaction. Finally, we have developed a social network topology generator that models mobile phone address books and the resulting contact graph used by propagating malware. Together, our environment is capable of simulating malware propagation on realistic network topologies, capacity constraints, and address book contact graphs among millions of mobile phones.

In the remainder of this paper, we first discuss related work in

mobile malware propagation in Section 2. Section 3 then describes our simulation methodology, network model, and contact graph models we use to explore malware propagation in mobile phone networks. Section 4 presents results of simulating various propagation scenarios using VoIP and MMS. Finally, Section 5 concludes the paper by discussing the implications of our results on defending provider networks against propagating malware.

2. RELATED WORK

The devastating outbreaks of Internet worms and viruses since the turn of the century inevitably led to the widespread investigation of malware propagation on the Internet (e.g., [10, 19, 23, 34]). Malware propagation on mobile phone networks, however, has received only limited attention, presumably due to few reports of extensive cell phone malware outbreaks. Rather than wait to react to widespread outbreaks, though, we believe it is important to investigate their potential effects as a basis for proactively defending against them.

Most previous work on mobile phone malware propagation has focused on Bluetooth worms, such as Cabir [7] and CommWarrior [8], in which infected devices discover and infect victim devices based on physical proximity. Su et al. [26] went to various locations and measured Bluetooth usage and duration of contact, finding that half of the phones encountered were in sufficiently long contact for malware to transfer itself. Kostakos et al. [12] deployed Bluetooth monitoring equipment in downtown Bath, England, and found that only 8% of their users had discoverable Bluetooth devices, greatly limiting infection possibilities. Mickens and Noble [15] modified traditional analytic models to create a probabilistic queuing technique that accounted for movement and traffic patterns over various time durations. Zheng et al. [32] focused on modeling population distribution density, Bluetooth radius, and node velocity. Their results point to a variety of quarantine methods that could greatly reduce the virulence potential.

In contrast to proximity-based contact worms, malware propagating through the communication network has the potential to spread more quickly, infect more devices, and cause more substantial disruption of the network infrastructure. Smartphone environments suffer similar vulnerabilities as Internet hosts. For instance, Wang describes potential security exploits in the Symbian OS that malware can take advantage of [28], and Mulliner et al. describe a proof-of-concept buffer overflow exploit for a PocketPC GSM/WiFi smartphone [20]. Further, the market relies predominantly on a single operating system, shadowing the homogeneity of Windows on the Internet; as of the first quarter of 2007, 72% of smartphones ran Symbian OSes, with Linux and Windows a distant second and third [27]. And smartphones are becoming increasingly prevalent. Although smartphones are currently a small fraction of the market, predictions estimate 365 million smartphones will comprise 20% of the market within five years [1].

Instances of malware exist that foreshadow the potential of using the mobile phone network for propagation and causing damage. For example, CommWarrior can propagate via MMS as well as Bluetooth [8], and the RedBrowser (J2ME) trojan [6] sends costly SMS messages expensed to a customer's account.

Investigating defenses to such malware is now gaining attention. Bose and Shin [4] propose a behavioral approach to detect anomalous activity and initiation containment of malware propagating via MMS/SMS. Van Ruitenbeek et al. also investigate propagation of MMS/SMS malware and various responses [22], although within only a small user population with an unconstrained messaging server. Bose and Shin [3] further model malware propagating through both MMS/SMS and Bluetooth vectors, parameterizing their analysis based upon real messaging workloads from a large mobile provider (although malware would likely propagate more aggressively and extensively than profiles of normal traffic workloads).

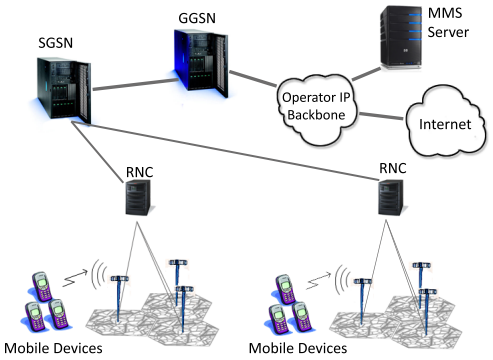

Figure 1: A simplified UMTS network

3. METHODOLOGY

We developed an event-driven simulator to investigate malware propagating using MMS and Voice over IP (VoIP) services. Our custom implementation gave us flexibility to create a scalable simulator that better served our specialized needs in terms of memory usage, performance and modeling. The simulator incorporates two important topology models that affect how malware spreads in such networks. First, it uses a model of the physical network topology that defines how mobile phone networks are connected and provisioned. The network topology fundamentally determines the physical constraints for malware propagation. Second, it uses a model of cell phone address books that, combined, create a contact graph topology among cell phone users. This contact graph determines how malware selects targets for infection. In the rest of this section, we describe how we represent and generate both of these topologies.[1]

3.1 UMTS network topology generator

We developed a configurable topology generator for Universal Mobile Telecommunications System (UMTS) networks, a third-generation mobile phone system standardized as the successor to the Global System for Mobile Communications (GSM). We first present a simplified overview of such a network as background, and then describe the topology generator we developed for studying malware propagation.

3.1.1 UMTS

UMTS defines a hierarchical set of nodes that interact to provide seamless voice and data services to handheld devices. Figure 1 illustrates a simplified version of a UMTS network for one carrier. We have assumed all communication is packet-based and that signaling and control channel effects are not significant when compared to packetized bandwidth limits. In the system, a mobile device connects through a radio interface to a *Node B* (essentially a radio tower). The Node B transmits and receives radio signals between the mobile devices. The network typically imposes bandwidth limits for each user, as well as the total bandwidth for radio cell/sector (thus also for the Node B). The Node B then for-

[1]A more extensive description can be found in a technical report [9].

wards data to a *radio network controller* (RNC) responsible for radio resource management, including hand-over and admission control. The RNC connects to a *Serving GPRS Support Node* (SGSN), which handles routing, authentication and charging functionality for a region. Typically, there are very few SGSNs in a network compared to the number of Node Bs. For clarity, we have not shown the mobile core connecting multiple SGSNs in a routable network, although we do model its effects. The hierarchy depicted in the figure is the path followed by packets in our simulator. We also show a *Gateway GPRS Support Node* (GGSN), which connects the mobile network to external networks, including the Internet. We do not model GGSN nodes since the scenarios we examine in our simulator focus primarily on communication within the mobile network. Finally, we show a Multimedia Messaging System (MMS) server that is connected through the operator backbone to the other elements. Typically, few MMS servers exist even for a very large region. In our simulator, we have simplified MMS communication by including only one MMS server connected directly to one of the SGSNs.

3.1.2 Simulated topology

The topology we used in our experiments was based on a population grid of 100x100 one square-mile cells centered around the Boston metropolitan area derived from U.S. census data. It contained 7,234,667 people initially, scaled down in each cell according to U.S. cell phone penetration (78%) [16], resulting in a total of 5,621,336 cell users. As a simplifying assumption, we did not model the mobility of cell phone users because the topology constraints overshadowed the importance of mobility in our scenarios. For both malware propagating through VoIP and MMS, the bottlenecks in the network occur up the network hierarchy or in central servers and as a result we do not expect mobility to have much of an effect.

We used the population grid as input into a topology generator to create a Radio Access Network (RAN) and packet-switched Core Network (CN) for a hypothetical operator servicing the region. Our topology generator attempts, very coarsely, to mimic some aspects of the network design process. In reality providers balance a wide range of factors — land-use and population data, distributions of subscribers and servers, radio propagation issues, etc. — to ensure coverage and capacity while minimizing system cost. In our case, we simply target our model to meet the capacity demands of the population base in the region. Based on the selected malware propagation vectors (MMS and VoIP calls), we focus solely on the packet-switched domain and omit the signaling involved (invoking nodes in the circuit-switched domain).

The topology generator uses a bottom-up strategy to construct the network. It first places the radio cells and Node Bs (radio base stations), and then constructs the (fixed) transport network from the radio access edge to the core network. It adds network nodes by connecting them to previously added nodes as it generates each layer of the hierarchy, while obeying certain key capacity constraints:

1. *Place radio cells and Node Bs.* Radio cells were placed one per grid cell to ensure that the maximum transmission range was not exceeded. For densely populated regions, like city centers, the radio cells would typically be smaller than our square mile grid cells in order to meet realistic traffic demands. Unfortunately, the coarseness of the available population density data makes it difficult to do this in a meaningful way. For future work we plan to investigate the possibility of combining it with land use data to solve this problem.

2. *Add RNCs by grouping a number of adjacent Node Bs together and connecting them to an RNC.* Up to 256 Node Bs in a 16 by 16 grid cell area were connected to an RNC. A maximum of 256 was chosen as a conservative assumption that is an even multiple of 4, to match requirements of the model construction algorithm that builds quadratic regions. In practice, RNCs vary in capacity and some can handle several hundred Node Bs. However, in a real dimensioning case, one would also need to consider other aspects and include further constraints beyond the mere number of Node Bs.

3. *Add SGSNs by grouping a number of adjacent RNCs together and connecting them to an SGSN.* The number of simultaneously attached users of an SGSN was limited to 10,000, and we assumed that all users may be attached simultaneously. We chose 10,000 simultaneously attached users as a conservative assumption for this SGSN performance metric, as advertised maximum capacity for SGSNs can be an order of magnitude higher. On the other hand, other constraints related, for instance, to bandwidth should also be considered; but doing so requires first estimating typical traffic characteristics, and we lacked meaningful data on which to base such estimates.

4. *Interconnect SGSNs to form a core network.* Lacking any significant empirical data regarding the preferred topologies for the core network, we chose to use a model similar to the Waxman model [29], i.e., a distance-biased random topology, where the probability of connecting two nodes is inversely proportional to the distance.

5. *Create an MMS server node and attach it to the network.* For simplicity, we connect the MMS server to an SGSN. Providers currently provision the MMS message forwarding server with a relatively low message rate, such as 100 messages/sec, to service millions of users during peak hours [2]. Hence, we assume a single server with a capacity of 100 messages/sec as the baseline case. The expected message load per user per busy hour can vary significantly from operator to operator, depending on pricing model. The assumption made here is for a message rate based on charges per message. We also simulated scenarios where the MMS server capacity was several times larger. However, the results are qualitatively the same as the MMS server will not be dimensioned to handle users behaving like an aggressive worm (i.e., sending large numbers of messages as quickly as possible).

For our scenario, the resulting topology consisted of 9,616 Node Bs, 49 RNCs, 49 SGSNs and 1 MMS server. We coarsely approximated link bandwidths by assuming E1s (2 Mbps) connecting a Node B and RNC, fast Ethernet (100 Mbps) connecting a RNC and SGSN, and Gigabit Ethernet (1 Gbps) links between SGSNs. The granularity of our Node B placement was a limiting factor of our initial population data. A finer granularity would, no doubt, offer a more detailed and accurate picture malware propagation.

3.2 Contact graph topology

When self-propagating malware infects a node, it needs to determine which nodes to contact to propagate the infection. Internet worms have typically used random scanning, in which an infected node generates random IP addresses (perhaps with a bias) as targets. The equivalent for our scenario would be randomly generating phone numbers to dial or send MMS messages. Given the relatively slow contact rate for mobile malware (e.g., one probe every

two seconds for mobile malware vs. 4,000 probes/sec for Slammer [18]), this method is not nearly as effective on mobile phone networks. Instead, as with email viruses, cell phones have address books that malware can use to determine which phones to contact and attempt to infect. Thus, modeling the distribution of contacts, as well as who the contacts are, form important bases for modeling malware propagation in mobile phone networks.

3.2.1 Address book degree distributions

We did not find published studies of cell phone address book characterizations. Lacking published data, we decided to consider three distributions for modeling the number of contacts in cell phone address books based upon previous work in related areas and a small experiment we undertook.

First, we consider a power-law distribution based on previous work modeling email address book distributions for studying email viruses. For example, Newman et al. [21] obtained address book data from a large institution and observed it to be heavy-tailed. Zou et al. [33] collected data from the size of Yahoo! email groups and found that the distribution is well modeled by a power law.

Second, we consider a log-normal distribution based on a study of an online social network. Liben-Nowell [13] studied the "friends" of LiveJournal.com users, counting the in and out degrees of each user. Since the Web site required active participation and manual addition of friends, he reasoned it was a valid indicator of real-world social networks. The data showed a long-tailed distribution best characterized by a log-normal distribution.

Finally, we also consider an Erlang distribution [30] based on the results of our experiment. After reviewing the literature, we were concerned that scale-free distributions would not accurately represent a cell phone social network topology. For one, cell phones limit the number of contacts; for example, a modern phone like the LG enV has a phone book capacity of 1,000 [17]. Furthermore, it seems unlikely that the large majority of users would have only two or three contacts that would result from such a distribution.

As a small experiment, we solicited the UCSD CSE department and Ericsson Research in Sweden, asking people to report the number of contacts in their phone address books. With 73 responses, we found that an Erlang distribution provided a good fit for the data (although admittedly the number of samples is too small to make definitive conclusions). An Erlang probability distribution is characterized by the formula:

$$P(x) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{(k-1)!}$$

where λ provides the rate of change and k defines the shape of the curve. We found that if we set $\lambda = .04$ and $k = 3$, we could create a distribution with an overall average of 65 contacts, which matched our anecdotal evidence as well as survey data.

In our experiments, we evaluated networks with different degree distributions — log-normal, power law and Erlang — ranging from 1 to 1,000 contacts. For a log-normal distribution, we modeled our distribution based on Liben-Nowell's LiveJournal.com data set. For the power law distribution, we used a power exponent of -1.7, the value reported by Zou et al.'s Yahoo study. Unfortunately, without more data available on cell phone address books, it is difficult to know the correct topology model.

3.2.2 Node attachment

To complete the generation of a contact graph topology, we also need to model how phones connect to each other in addition to their degree distributions. We use a model that assumes that mobile phone users are more likely to communicate with people they are in regular physical contact with, and as a result the contacts in an address book will more likely be geographically nearby than distant. We again incorporate results from Liben-Nowell et al. [14]. They correlated LiveJournal.com users with their geographic location based on zip code and compared that information to the location of their friends. In contrast to previous work, they found that the probability that two users were "friends" was proportional to the inverse of the number of people between them. Thus, when determining if two nodes are connected, the actual geographic distance is not as important as the size of the population that separates two nodes. Their results, based on U.S. population patterns, provide a more accurate picture of a social network. However, they also discovered this rule only held for about 70% of the relationships. The remaining 30% could not be characterized by the measure.

We make use of both of these findings in our simulator. After randomly assigning a phone's address book contact size, we assign 70% of its contacts probabilistically based on the population between cells. We then assign the remaining 30% of the contacts uniformly randomly within the constraints of the size of the address book of each phone.

4. EXPERIMENTAL RESULTS

In this section we present the results of various experiments of malware propagation on mobile phone networks using our simulator. Each simulation used a one-second time step, and we simulate for 12 hours to explore infection behavior. All runs start with one infected phone. As described above, we use the UMTS generator to create a network servicing the equivalent of the Boston metropolitan area.

We begin by examining how VoIP and MMS malware spread given the constraints of the network and address book topology. We then briefly explore a few techniques that malware authors could use to increase infection rate. In general, we show our results as the percentage of the total population infected as a function of malware propagation time. Each point is the average of five runs (we found little variation). More extensive results appear in our technical report [9]. In our experiments, we did not model actual customer traffic on the network, only malware traffic. We made this simplifying assumption because, once malware that exploits a widespread vulnerability starts to propagate, it will dominate traffic on the network.

4.1 VoIP scenario

The Voice over IP scenario models malware that exploits software dedicated to handling voice data. To propagate, it selects and dials a phone number. When that phone answers, it sends the payload over the channel and repeats. The victim phone becomes infected after the entire payload has been delivered, which is dependent upon latency and capacity constraints. It resembles Internet worms in that a user can do little to stop an infection once it has begun communication with a user. We assume that all phones are vulnerable to the VoIP attack, which represents the worst-case scenario. We recognize a real attack would likely only affect one phone type, and hence only a fraction of the population. We have simulated scenarios with only a fraction of the population vulnerable with qualitatively similar results.

We first look at an unconstrained environment without bandwidth or capacity limits. Figure 2 shows the rate at which malware would spread through the network infrastructure given various address book topologies. The "complete" case emulates an address book that contains every contact in the population. It results in the fastest propagation, infecting 90% of the population after 86 seconds. The address book contact graph also has a substantial effect on propagation. The log-normal and power law cases asymp-

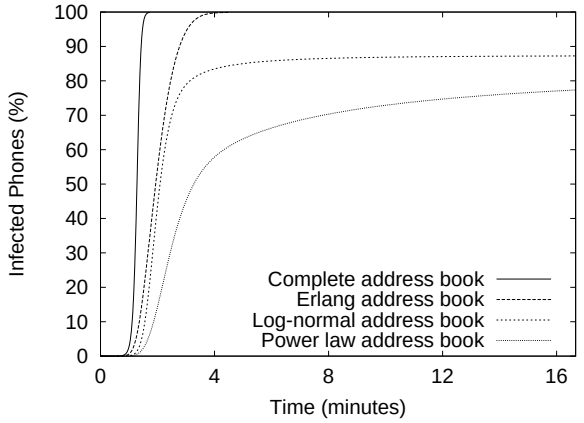

Figure 2: VoIP infection levels with unconstrained bandwidth propagation.

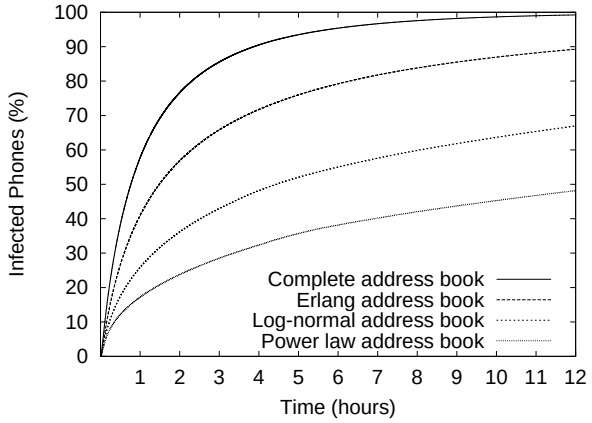

Figure 3: VoIP infection levels with constrained bandwidth propagation.

tote because the topologies create a disconnected social network. The Erlang topology is rich enough to consistently generate a fully-connected graph and reaches 90% infected after 167 seconds, about twice as long as "complete".

Next we incorporate the bandwidth and capacity constraints determined by the generated network topology. Figure 3 shows malware propagation for the various address book topologies. On a realistic network the malware propagates orders of magnitude slower than when unconstrained, losing the characteristic "S"-curve. Even with a "complete" address book (a fully connected graph), propagation took 3.9 hours to infect 90% of the population, over 160 times as long as the unconstrained case. Malware propagation also varies according to the address book model, although the variation is not as pronounced as when there are no bandwidth constraints. At the time when propagation with a complete address book reaches 90% infection, malware propagating with address books modeled using the Erlang, log-normal, and power law distributions only reached 71.1%, 47.6%, and 31.9% of the population, respectively.

Whereas Internet worms like Code Red and Slammer infected a large proportion of vulnerable hosts before reaching bandwidth limitations [18], VoIP malware propagation in a mobile network becomes bandwidth-limited very early in the propagation. One might suspect that bandwidth at the radio cell level presents the most significant hurdle. However, measuring average congestion across the network hierarchy in the model points to the RNC to SGSN link as the primary bottleneck. The precise location of bottlenecks will depend greatly on the assumptions made for dimensioning of the system. Thus, they are likely to vary somewhat from operator to operator.

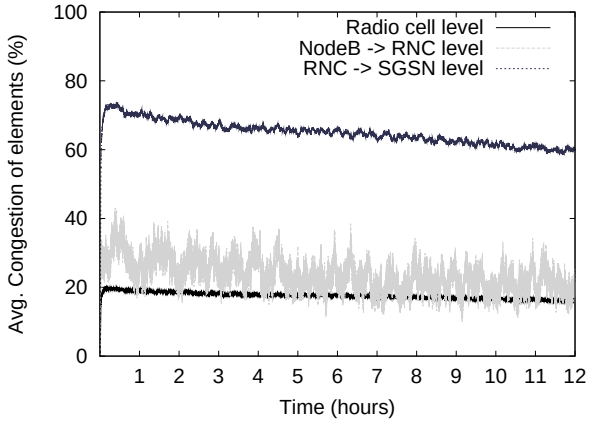

Figure 4: Average congestion at different network levels during VoIP propagation.

Figure 4 shows the average congestion when using the Erlang address book topology for the radio cell level, the Node B to RNC links, and the RNC to SGSN links. We compute average congestion by summing the amount of bandwidth used across all elements at each link and dividing by the total amount of bandwidth available. The measurement provides a rough network-wide gauge of congestion. Because of the large fan-in ratio from Node Bs to RNCs, a large flow of traffic constantly arrives at each RNC. The RNC then has to forward this traffic to a SGSN. However, the total traffic from the Node Bs overwhelms the total bandwidth available to the RNC. Interestingly, the congestion peaks very early, within approximately 5–7 minutes. Then congestion decreases slightly over time even though more phones become infected. This counter-intuitive result is a side-effect of how we model congestion. As phones finish enumerating their address books, they begin to randomly dial numbers, which in our simulation does not add to overall data congestion since these attempts are likely to fail in making end-to-end connections.

4.2 MMS scenario

The Multimedia Messaging Subsystem (MMS) is a service for sending messages with attachments, such as photos or videos. If malware were to spread through the use of MMS messages, then a message would be generated on the infected phone and routed to a centralized MMS server. This server often has a severe capacity constraint (Section 3.1.1) which limits the total number of messages per second sent or received. The target phone then downloads the message from the server when a user accesses it. To model user wait times for accessing messages, we employed a mixture of two Gaussian distributions centered at 20 seconds and 45 minutes.

Figure 5 shows the propagation characteristics for this MMS scenario for the different address book topologies. Compared to VoIP, MMS malware takes dramatically longer to infect the same number of devices due to the centralized MMS server bottleneck. Recall that we provisioned the message server with a limit 100 messages/sec, so MMS senders and receivers equally sharing the capacity result in a maximum achievable rate of 50 infections/sec.

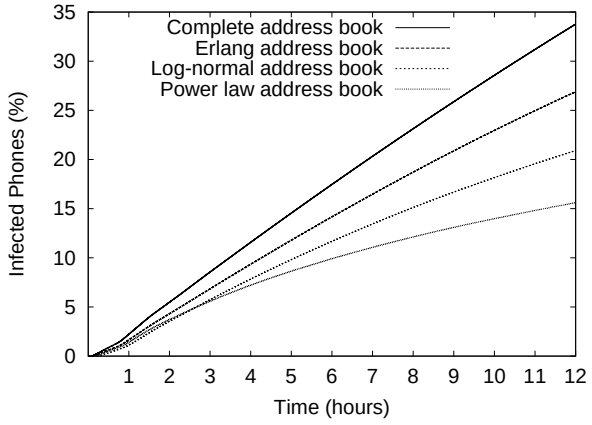

Figure 5: Infection levels for the MMS scenario with different address book topologies.

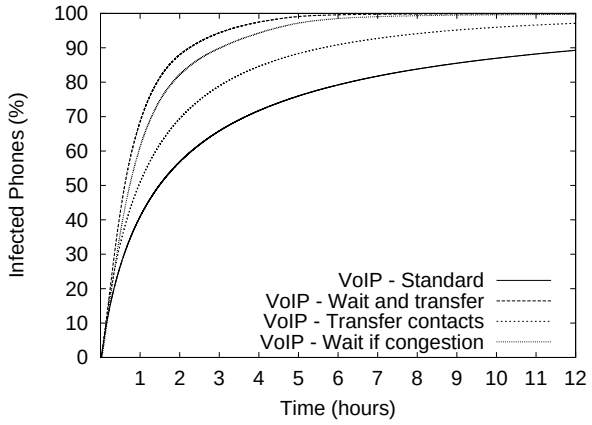

Figure 6: Infection level for VoIP malware that avoids congestion and transfers contacts.

The results show that the address book topology does have an effect on the rate at which malware can spread, but the differences are less pronounced than with VoIP. The slight bump at 45 minutes corresponds to the second mean of the bimodal wait time distribution, which is when the malware achieves the full rate of infection. We also experimented with removing the wait time altogether as a worst-case scenario. However, removing the wait time only improves propagation time by 3%, indicating that wait time is not a dominant factor.

Malware like CommWarrior [8] uses messages, such as "Game from me. It is FREE !", that require user intervention before the phone becomes infected. We also modeled MMS propagation assuming that only a fraction F of users act on the message. Because of the centralized server bottleneck, the malware propagates roughly a factor F times as slow as the automatic MMS scenario.

4.3 More sophisticated malware

Internet malware has the potential to use various mechanisms to accelerate infection, such as using hit lists, dividing the known address space, etc. [25]. In a mobile phone network, an attacker can also try to leverage knowledge of the constraints of cell phone networks to engineer malware that can spread faster.

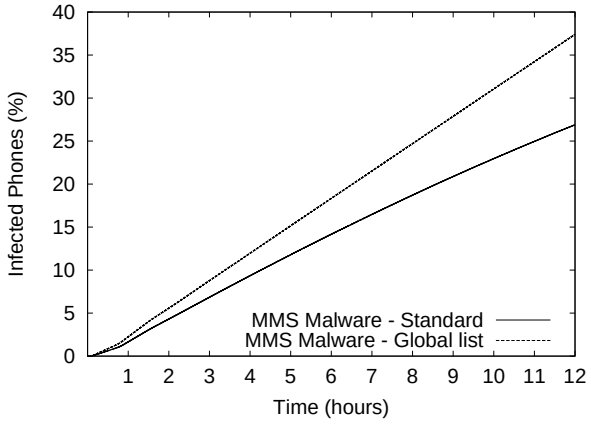

Figure 7: Infection levels for MMS malware that uses a centralized list to coordinate infections.

Clever malware will want to avoid congestion and distribute the work required to contact phones. As one design point, we evaluate malware that backs off when detecting congestion by sleeping for 10 seconds. At the same time, the malware divides and evenly distributes contacts from its address book when one phone infects another. For the simulations on more aggressive malware, we present results for only one address book topology, the Erlang topology. We present results using this topology because it does not have the unrealistic bias towards very low degree address books. Figure 6 demonstrates the effects of these techniques for VoIP malware. Distributing contacts further improves propagation, although not as substantially as avoiding congestion. The MMS scenario benefits very little from these techniques due to the capacity constraints of the centralized message server.

MMS-based malware might be attractive to attackers since both phones do not need to be powered on at the same time. However, it is clear that the capacity of the MMS server severely limits propagation. To optimize propagation, malware could perhaps coordinate propagation by sharing address books and eliminating duplicate effort. As an extreme of this approach, we model a scenario where malware uploads address books to a central server on the Internet to create a complete, global contact list. Figure 7 shows the potential of this optimization. Although it does not approach the speeds of VoIP malware spreads, it reaches almost half the population after only twelve hours. Further, its average rate of infection when is 48 new infections per second (compared to 35 in the standard MMS scenario), nearly optimal considering the provisioned 50 messages/sec capacity.

5. IMPLICATIONS FOR DEFENSE

What do these results suggest about defending cell networks against aggressive malware? In concluding, we discuss the implications on defense. Due to how network providers provision mobile phone networks, aggressive malware very quickly bottlenecks capacity on internal links. As a result, the malware quickly launches an effective denial-of-service attack, preventing other customers from using a service (VoIP or MMS). Such a denial-of-service would likely lead to unhappy customers, public relations disasters, and a direct loss of revenue for providers.

At the same time, the resulting bandwidth bottlenecks are clear signals to operators that the network has a problem, and these signals occur very early in the infection. As a result, even if a network provider cannot prevent malware propagation in the first place, it

still has an opportunity to react before malware infects most of the vulnerable population.

Borrowing defense techniques developed for Internet worms, mobile phone providers could employ defenses such as rate limiting [31] or containment via blacklisting and content filtering [19]. When simulating rate limiting, in which an operator restricts calls to every M minutes after detecting an outbreak, our results indicate that rate limiting in fact accelerates malware. For the same reasons as congestion avoidance accelerates malware, rate limiting is effectively congestion avoidance implemented by the operators instead of the malware.

Containment using blacklisting would quarantine phones after suspicious behavior (e.g., random VoIP dialing or persistent messaging). When simulating the use of blacklisting, though, even with an aggressive policy of blacklisting users a large number of devices become infected, albeit more slowly. Moreover, removing users from the network again reduces congestion and allows non-blacklisted infected phones to continue to spread.

Containment using content filtering [24] appears promising, however. Traffic in a mobile phone network is much more centrally controlled compared to the Internet; indeed, although the MMS server is a central bottleneck for the network, it is also an ideal place to analyze traffic, generate content signatures, and potentially push signatures down the hierarchy to halt propagation. Investigating whether such an approach is practical remains future work.

Finally, we also have not discussed how to remove infections from phones. Whether customers have to visit provider stores for patching, download patches to their computers and apply them via Bluetooth or USB, or use alternate communication services (e.g., MMS to remove VoIP infections), this problem also remains important future work.

Acknowledgments

We are grateful to our colleague Eva Fogelström for insightful comments and discussions. Johan Gard provided valuable information regarding system dimensioning. David Liben-Nowell was kind enough to provide the LiveJournal.com dataset. Finally, we would like to thank Kostas Anagnostakis and our anonymous reviewers for their time and insightful comments regarding this paper. Support for this work was provided in part by NSF CyberTrust Grant No. CNS-0433668, AFOSR MURI Contract F49620-02-1-0233, and the UCSD Center for Networked Systems.

6. REFERENCES

[1] BERGINSIGHT. Smartphone operating systems, May 2007. http://www.berginsight.com/ReportPDF/Summary/BI-SOS-SUM.pdf.

[2] BODIC, G. L. *Mobile Messaging: Technologies and Services, SMS, EMS and MMS*, 2nd ed. Wiley and Sons, 2005.

[3] BOSE, A., AND SHIN, K. On mobile viruses exploiting messaging and bluetooth services. In *Internat'l Conf. on Security and Privacy in Comm. Networks (SecureComm'06)* (Sept. 2006).

[4] BOSE, A., AND SHIN, K. Proactive security for mobile messaging networks. In *Proc. of ACM WiSe'06* (Sept. 2006).

[5] DAGON, D., MARTIN, T., AND STARNER, T. Mobile phones as computing devices: The viruses are coming! *IEEE Pervasive Computing 03*, 4 (Oct. 2004).

[6] F-SECURE. F-Secure Trojan Information Pages: RedBrowser. http://www.f-secure.com/v-descs/redbrowser_a.shtml.

[7] F-SECURE. F-Secure Virus Information Pages: Cabir. http://www.f-secure.com/v-descs/cabir.shtml.

[8] F-SECURE. F-Secure Virus Information Pages: Commwarrior. http://www.f-secure.com/v-descs/commwarrior.shtml.

[9] FLEIZACH, C. B. Can You Infect Me Now? A Treatise on the Propagation of Malware in a Cellular Phone Network. Tech. Rep. CS2007-0894, UCSD, June 2007.

[10] GANESH, A. J., MASSOULIE, L., AND TOWSLEY, D. The effect of network topology on the spread of epidemics. In *Proc. IEEE Infocom* (2005).

[11] HYPPONEN, M. Malware goes mobile. *Scientific American 295*, 5 (Nov. 2006).

[12] KOSTAKOS, V. Experiences with urban deployment of Bluetooth (given at UCSD), Mar. 2007. http://www.cs.bath.ac.uk/~vk/files/pres_ucsd.pdf.

[13] LIBEN-NOWELL, D. *An Algorithmic Approach to Social Networks*. PhD thesis, MIT, 2005.

[14] LIBEN-NOWELL, D., NOVAK, J., KUMAR, R., RAGHAVAN, P., AND TOMKINS, A. Geographic routing in social networks. *Proc. of the Nat'l Academy of Sciences 102*, 33 (Aug. 2005).

[15] MICKENS, J. W., AND NOBLE, B. D. Modeling epidemic spreading in mobile environments. In *Proc. of ACM WiSe'05* (Nov. 2005).

[16] MOBILE WORLD. The Mobile World Briefing, October 2006. http://www.themobileword.com/tmwdev.objects/documents/pdf/TMWBriefingIs%sue40.pdf.

[17] MOBILEDIA. LG enV (VX9900) Specifications. http://www.mobiledia.com/phones/lg/env.html.

[18] MOORE, D., PAXSON, V., SAVAGE, S., SHANNON, C., STANIFORD, S., AND WEAVER, N. Inside the Slammer Worm. *IEEE Security and Privacy 1*, 4 (July 2003), 33–39.

[19] MOORE, D., SHANNON, C., VOELKER, G. M., AND SAVAGE, S. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the 2003 IEEE Infocom Conference* (Apr. 2003).

[20] MULLINER, C., VIGNA, G., AND LEE, W. Using labeling to prevent cross-service attacks against smart phones. In *Proc. of DIMVA'06* (July 2006).

[21] NEWMAN, M., FORREST, S., AND BALTHROP, J. Email networks and the spread of computer viruses. *Phys. Rev.E 66*, 3 (Sept. 2002).

[22] RUITENBEEK, E. V., COURTNEY, T., SANDERS, W. H., AND STEVENS, F. Quantifying the Effectiveness of Mobile Phone Virus Response Mechanisms. In *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (June 2007), pp. 790–800.

[23] SERAZZI, G., AND ZANERO, S. Computer virus propagation models. In *Tutorials of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS'03)* (Oct. 2003).

[24] SINGH, S., ESTAN, C., VARGHESE, G., AND SAVAGE, S. Automated worm fingerprinting. In *Proceedings of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI)* (2004), pp. 45–60.

[25] STANIFORD, S., PAXSON, V., AND WEAVER, N. How to 0wn the Internet in your spare time. In *Proc. of USENIX Security '02* (Aug. 2002).

[26] SU, J., CHAN, K. K. W., MIKLAS, A. G., PO, K., AKHAVAN, A., SAROIU, S., DE LARA, E., AND GOEL, A. A preliminary investigation of worm infections in a Bluetooth environment. In *Proc. of ACM WORM'06* (Nov. 2006).

[27] SYMBIAN. Symbian fast facts, Mar. 2007. `http://www.symbian.com/about/fastfacts/fastfacts.html`.

[28] WANG, R. X. Symbian OS - Mysterious playground for new malware. *Virus Bulletin* (Sept. 2005).

[29] WAXMAN, B. M. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications* (Dec. 1988).

[30] WIKIPEDIA. Erlang distribution — Wikipedia, The Free Encyclopedia, 2007. [Online; accessed 7-April-2007].

[31] WONG, C., BIELSKI, S., STUDER, A., AND WANG, C. Empirical analysis of rate limiting mechanisms. In *Internat'l Symp. on RAID'05* (Sept. 2005).

[32] ZHENG, H., LI, D., AND GAO, Z. An epidemic model of mobile phone virus. In *Proc. of Internat'l SPCA'06* (Jan. 2006).

[33] ZOU, C., TOWSLEY, D., AND GONG, W. Email worm modeling and defense. In *Proc. of Internat'l Conf. on Computer Comm. and Networks* (Oct. 2004).

[34] ZOU, C. C., GONG, W., AND TOWSLEY, D. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM conference on Computer and communications security* (2002), pp. 138–147.