A Preliminary Investigation of Worm Infections in a Bluetooth Environment

Jing Su[†], Kelvin K. W. Chan[†], Andrew G. Miklas[†], Kenneth Po[⋆], Ali Akhavan[†]
Stefan Saroiu[†], Eyal de Lara[†], Ashvin Goel[⋆]

[†]Department of Computer Science, University of Toronto
[⋆]Department of Electrical and Computer Engineering, University of Toronto

ABSTRACT

Over the past year, there have been several reports of malicious code exploiting vulnerabilities in the Bluetooth protocol. While the research community has started to investigate a diverse set of Bluetooth security issues, little is known about the feasibility and the propagation dynamics of a worm in a Bluetooth environment. This paper is an initial attempt to remedy this situation.

We start by showing that the Bluetooth protocol design and implementation is large and complex. We gather traces and we use controlled experiments to investigate whether a large-scale Bluetooth worm outbreak is viable today. Our data shows that starting a Bluetooth worm infection is easy, once a vulnerability is discovered. Finally, we use trace-drive simulations to examine the propagation dynamics of Bluetooth worms. We find that Bluetooth worms can infect a large population of vulnerable devices relatively quickly, in just a few days.

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, Measurement

Keywords

worms, malware, Bluetooth

1. INTRODUCTION

In the span of a few years, Bluetooth has become one of the most popular wireless protocols. It has been forecasted that in a few years, Bluetooth-enabled devices will outnumber Wi-Fi devices five to one, with over 77% of cell-phones, 60% of PDAs, and 67% of notebooks having built-in Bluetooth radios [27]. Recently, car manufacturers have started to equip automobiles with Bluetooth, linking cell-phones to the cars' audio systems. Even consumer appliance manufacturers have started to incorporate Bluetooth radios in microwaves, refrigerators, and washing machines [26].

Bluetooth is complex: its current Linux implementation consists of over 25K lines of kernel code. Because Bluetooth's codebase is large, we believe bugs are likely to be present in current implementations. In fact, over the past year, there have been several reports of malicious code exploiting vulnerabilities in the Bluetooth protocol [18, 5], including attempts to create a worm infection [8, 16]. While no large-scale Bluetooth security attacks have been reported, a worm propagating over the Bluetooth protocol could cause massive disruptions with serious consequences. A malicious program could launch a DoS attack and bring down a segment of the cellular network. Cell-phone spyware could collect personal information. The consequences of malicious programs controlling car components could be drastic.

While the research community has started to investigate a diverse set of Bluetooth security issues [17, 10, 30, 28, 15], little is known about the feasibility and the propagation dynamics of Bluetooth worm infections. This paper is an initial attempt to remedy this situation.

First, we investigate whether a large-scale Bluetooth worm outbreak is viable in practice. Even if a program can exploit a Bluetooth vulnerability to replicate itself, a large-scale outbreak might never develop. If vulnerable devices are few and far between, a worm might never reach many victims. In this case, the threat of a large-scale Bluetooth worm infection is minimal.

We conducted controlled experiments and we gathered traces of Bluetooth activity in different urban environments to determine the feasibility of a worm infection. We find that Bluetooth-enabled devices are prevalent and that the device population is relatively homogeneous. This suggests that a worm exploiting a vulnerability in a popular Bluetooth implementation could spread quickly. We also find that devices typically remain within the range of our Bluetooth radios long enough for a potential infection to occur. Finally, we find that walking cannot prevent a person's device from becoming infected: an infection can occur even when two people carrying Bluetooth devices are walking in opposite directions. All these results suggest that a large-scale Bluetooth worm outbreak is viable today.

Second, we use trace-driven simulations to examine the propagation dynamics of a Bluetooth worm in a large population. Our simulations reveal that Bluetooth worms can infect many devices relatively quickly, in just a few days. We find that a worm's infection rate is not strongly affected by how many devices are infected first: the infection propagates quickly whether it started by infecting 100 or 1000 devices. Instead, we find that the worm outbreak's start time is more important: an infection starting on a week-end or

WORM'06, November 3, 2006, Alexandria, Virginia, USA.

during an off-peak hour (e.g., late evening) spreads more slowly.

While preliminary, our findings have a few implications. First, although Bluetooth worms spread orders of magnitude more slowly than Internet worms, Bluetooth worms spread quickly enough that human-mediated counter-response will be difficult to implement in practice. Second, defense solutions based on geographical locality could be adequate for Bluetooth worms. For example, placing monitoring points in high-traffic locations, such as airports, could prevent a large-scale worm outbreak. Section 7 discusses our implications in more detail.

The rest of the paper is organized as follows. Section 2 presents a brief Bluetooth primer. In Section 3, we discuss how Bluetooth worms are different from other types of worms. In Section 4, we describe the diversity of known security attacks on the Bluetooth protocol. In Section 5, we investigate the feasibility of a large-scale Bluetooth worm outbreak today. In Section 6, we use simulations to examine the propagation dynamics of Bluetooth worms. Finally, in Section 7 we present our conclusions and we briefly discuss the implications of our findings.

2. A BRIEF BLUETOOTH PRIMER

Bluetooth [3] is a communication protocol for low-power, wireless devices operating in the unlicensed band at 2.4GHz. While the Bluetooth specification makes provision for ranges of up to 100 meters, most of the radios of today's mobile devices have ranges of 10-20 meters.

There are two ways in which a device can initiate a Bluetooth connection: (1) by directly contacting the address of another device, or (2) by broadcasting "inquiry" messages to discover other devices. Most of today's Bluetooth devices provide the user with the option to make them discoverable. Upon receiving an "inquiry" message, a discoverable Bluetooth device will reply with an answer that includes its user-configurable device name and its device type (e.g., a cell-phone or a laptop). In our experiments, we used the "inquiry" mechanism to discover other nearby devices. Because Bluetooth devices can be set to be non-discoverable, our trace-based estimates on the prevalence of Bluetooth devices are conservative. However, there are well-known brute-force ways to discover non-discoverable Bluetooth devices [30].

Bluetooth supports two types of link-layer connections: a synchronous connection-oriented (SCO) link, where no lost packets are retransmitted, and an asynchronous communication link (ACL) where packet retransmission is applied. Once a link is formed, data can be exchanged using a socket-based interface in a manner similar to Internet-based protocols. Setting-up a Bluetooth connection is time-consuming; in fact, for short transfers, the time spent setting up the connection dominates the transfer times. This indicates that the connection setup time affects a Bluetooth worm infection rate. Section 5 provides quantitative data on the throughput and setup connections times in Bluetooth.

When communicating, two devices can use cryptographic protocols to create a shared key. With this key, they can encrypt all data exchanged between them. This key is also used in subsequent re-connections; in this way, device re-authentication is no longer necessary. In Bluetooth, creating a shared key is called "pairing" two devices. To generate the shared key, Bluetooth uses a per-connection unique PIN: a number having four to seven digits.

3. BLUETOOTH WORMS ARE DIFFERENT THAN OTHER WORMS

This section describes how the characteristics of Bluetooth worms are different than those of other classes of worms, such as Internet worms and worms propagating in mobile ad-hoc networks (MANETS.)

The way in which a worm infection propagates in a Bluetooth environment is likely to be different than the spread of an Internet worm [29, 23, 22]. On the Internet, a worm typically infects a well-provisioned PC over a fast, bandwidth-rich Internet connection. Once infected, an Internet host can choose to attack any other host with an Internet connection. In contrast, Bluetooth worms infect a different class of devices: mobile, power-constrained devices with Bluetooth radios. A Bluetooth infection occurs only when the infection source and the victim are located near each other, as most commodity Bluetooth radios only have a range of 10-20 meters. Unlike Internet worms, Bluetooth worms' propagation is driven by how the vulnerable devices interact, move, and travel.

At a first glance, Bluetooth worms are similar to worms propagating in MANETS [6, 7]: both types of worms propagate in mobile, power-constrained environments. However, worm infections over Bluetooth are likely to differ from worm infections in MANETS. First, unlike Bluetooth, MANETS typically support multi-hop routing: once infected, a node can attack any other node participating in the network. Second, the range of Bluetooth radios is approximately an order of magnitude smaller than the range of Wi-Fi radios. Third, there is a much wider variety of Bluetooth devices with different hardware and software characteristics than Wi-Fi devices. Finally, Bluetooth is a very popular protocol, deployed over millions of devices, whereas typical deployments of MANETS consist only of hundreds of nodes.

4. DIVERSITY OF BLUETOOTH SECURITY ATTACKS

In this section, we show that Bluetooth-enabled devices are already subject to sophisticated types of attacks. We start by investigating the complexity of the Bluetooth protocol's design and implementation. Next, we examine three classes of known security attacks.

Protocol Complexity. The Bluetooth protocol is complex. The document presenting the core specification of Bluetooth's latest version has over 1200 pages [2]. Table 1 shows the number of lines of code for supporting Bluetooth in Linux kernel version 2.6.15. There are over 25K lines of kernel code implementing the Bluetooth stack and drivers in Linux. The Bluetooth code size is about half the size of the TCP/IP stack implementation. We believe these figures suggest that errors and bugs are likely to be present in current Bluetooth implementations.

Cryptographic Vulnerabilities. Previous work has analyzed the cryptographic properties of the Bluetooth authentication mechanism [14, 12, 1, 20, 19, 28]. While these vulnerabilities are unlikely to lead to large-scale worm infections, these results indicate that Bluetooth's authentication is weaker than previously claimed.

Shaked and Wool have shown that it is relatively easy to crack the PIN used by a Bluetooth device when pairing with another device [28]. After finding the PIN, a malicious device can decrypt all communication exchanged between the paired devices or impersonate the victim device.

Social Engineering-based Attacks. A popular form of attack is to use a carefully chosen device name when pairing with the target device [5]. To complete the pairing process, the target device must ask for its user's permission while displaying the attacking device's name. A well chosen device name (e.g., "Your Friend" or

Bluetooth kernel stack	TCP/IP kernel stack	Bluetooth library	libc
25K (9K drivers)	50K	45K	526K

Table 1: The number of lines of code of the Bluetooth kernel and libraries in Linux kernel 2.6.15: The Bluetooth kernel stack (including drivers) is half the size of the TCP/IP kernel stack. To support Bluetooth, an application must be linked with a Bluetooth library, such as "bluez-utils" [4]. This library has 45K lines of code.

"Secret Admirer") could convince the user to authorize the pairing. This type of attacks is known by the term of "bluejacking".

More recently, there have been reports of a Bluetooth virus outbreak [8]. Cabir is a software program that repeatedly scans for nearby Bluetooth-enabled devices. Upon discovering a new device, Cabir transmits an installation file disguised as a security management utility. Once target users accept the incoming file, their devices become infected. Because it requires user intervention, Cabir has not been able to reach and infect a large device population. However, there are reports of Cabir-infected Bluetooth devices found in stores selling cell-phones and cell-phone accessories [8].

Attacks Exploiting Software Vulnerabilities. Several attacks exploiting Bluetooth implementation vulnerabilities have been reported [18]. In these attacks, a malicious device can gain access to data on a vulnerable device, issue AT modem commands, or establish a unauthorized "pairing" relationship. A previous study has measured the prevalence of some of these software vulnerabilities in a trace of Bluetooth-enabled phones captured at CeBIT 2004, a large IT exhibition taking place in Hanover, Germany [13]. Their trace has captured 1,269 discoverable Bluetooth devices over a period of four days. This study found that many devices (i.e. between 6% and 33% depending on the phone type) exhibit exploitable software vulnerabilities. This software vulnerability allowed the authors to retrieve the Bluetooth devices' address books.

Summary

While no large-scale Bluetooth security attacks have been reported, a diverse set of known security vulnerabilities already exists. Their presence coupled with the complexity of the Bluetooth specification and its large codebase size make us believe that Bluetooth-enabled devices will likely contend with increasingly complex attacks in the future, such as worms.

5. FEASIBILITY OF A BLUETOOTH WORM INFECTION

Even if a worm could exploit a security vulnerability in the Bluetooth protocol to replicate itself, a large-scale Bluetooth worm outbreak might never develop. If vulnerable Bluetooth devices are few and far between, and most inter-device contacts are short, a worm might never reach many victims. In this case, the threat of a large-scale Bluetooth worm infection is minimal.

In this section, we examine whether a large-scale Bluetooth worm outbreak is viable in practice. For this, we collected traces of Bluetooth activity and conducted controlled experiments in a Bluetooth environment. We use this data to answer four key questions:

1. Are discoverable Bluetooth-enabled devices prevalent today? While current figures estimate that 10% of today's cell-phones have Bluetooth radios [24], it is less clear whether Bluetooth devices with discoverable radios are widespread. If discoverable Bluetooth devices are prevalent, a worm can find and infect many devices, once released.

2. How heterogeneous is the population of devices? A heterogeneous population is more resistant to a worm exploiting a single software vulnerability.

3. Are typical inter-device contact durations long enough to allow a worm to replicate? A worm infection occurs only when an infection source is in contact with the target long enough for the worm to exploit the vulnerability and replicate itself. Even if a worm outbreak occurs, short inter-device contacts could slow the worm's infection rate.

4. Can a worm replicate between two devices moving in opposite directions at human walking speeds? The relative speed between two people walking could be too high for a worm to replicate. In this case, a mobile source is less "effective" because it could "miss" all vulnerable devices moving in opposite direction.

5.1 Methodology of Our Experiments

We collected three different traces of Bluetooth activity. Two of our traces are gathered inside Pacific Mall and Eaton Centre, two malls in Toronto, Canada. We gathered the third trace while riding the Toronto subway system. These three locations provide a broad coverage of different density and mobility characteristics one might find in various urban destinations.

We used Palm Tungsten T PDAs having 16MB of RAM with PalmOS version 5.0 to scan for Bluetooth devices. The Bluetooth radios of our PDAs are similar to the ones found in most commodity cell-phones: our empirical tests found that our PDAs' ranges are about 10 meters in an urban environment corresponding to the specifications presented on Palm's website [25]. Because a Bluetooth inquiry is a power-intensive procedure, we used a total of eight scanners. Each device sends "inquiries" over its Bluetooth interface. Our inquiry rate is variable: we increase it when no devices are discovered, and we decrease it when others answer our probes. We issue inquiries at least once every 10 seconds but never more often than once every 3 seconds. This variable rate deals with congestion scenarios when several devices answer simultaneously.

When collecting these traces, we had a behavior compatible to the environment we were scanning. For example, we were casually walking in the malls, we stopped briefly by their food courts, and we stood still while riding the subway. In this way, our data illustrates a scenario where an attacker behaves inconspicuously while launching a Bluetooth worm. We used two devices scanning simultaneously to collect the Eaton Centre and the Subway traces. We used only one device to collect the Pacific Mall trace.

We also conducted controlled experiments to determine whether walking can prevent a person's device from becoming infected. We placed one device on a wall at a T-junction hallway, while a person carried another device pacing themselves at a constant speed. The mobile device first issued inquiry requests. Once the stationary device is discovered, the mobile device transmitted a file. We performed several experiments. We set the size of the file at 500 bytes and at 25KB. We moved the mobile device at a speed of 1 m/s, corresponding to a typical walking speed, and 2 m/s, to approximate the relative speed of two people walking in opposite directions. Each experiment is repeated five times. Figure 1 illustrates the topology of our experimental setting. We chose the T-junction hallway because it combines both line-of-sight and obstructed inter-device transmissions.

Traces	*Pacific Mall Trace*	*Eaton Centre Trace*	*Subway Trace*
Start Time	11/26/2005 13:23:41 (Saturday)	11/16/2005 10:42:56 (Wednesday)	11/16/2005 06:49:37 (Wednesday)
End Time	11/26/2005 15:10:51 (Saturday)	11/16/2005 14:06:33 (Wednesday)	11/16/2005 20:32:10 (Wednesday)
Duration	107 minutes	204 minutes	Intermittent Trace
Total # of Devices Found	90	100	106
# of Cell-Phones	87 (97%)	84 (84%)	96 (91%)
# of Computers	3 (3%)	15 (15%)	9 (8%)
# of Headsets	0	1 (1%)	0
# of Unknown Device Types	0	0	1 (1%)

Table 2: Summary of our traces: We use two devices scanning simultaneously to collect the Eaton Centre and the Subway traces. We use only one device to collect the Pacific Mall trace. All times are EST.

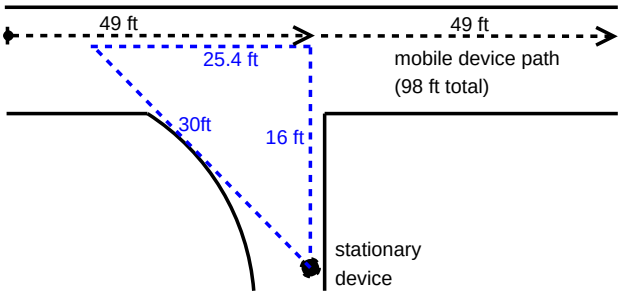

Figure 1: Topology of our controlled experiment.

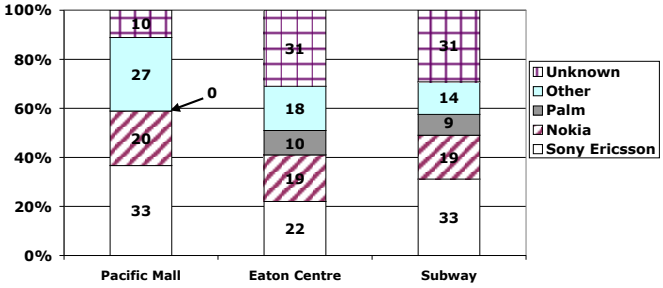

Figure 2: Breakdown of devices by their manufacturer: There are 11 manufacturers in the Pacific Mall, 13 in Eaton Centre, and 11 in the subway trace. Sony Ericsson and Nokia together account for 51% of all devices found.

5.2 Discoverable Bluetooth Devices are Prevalent Today

A high prevalence of Bluetooth devices facilitates the spread of a worm infection. This is analogous to the spread of human diseases: large-scale epidemics spread more quickly in densely populated environments.

Table 2 shows the number of Bluetooth devices discovered in our traces. Overall we discovered 288 devices: 90 and 100 devices inside the two malls, and 106 while riding the subway. Some devices were encountered in multiple traces. On average, we found one device every 15 seconds while tracing in the malls. The longest time we spent scanning without discovering any new device was 13 minutes inside Eaton Centre and 26 minutes inside Pacific Mall. These findings show that discoverable Bluetooth-enabled devices are already prevalent in urban environments.

5.3 The Population of Devices is Homogeneous

Worm infections occur only when target devices have the same vulnerability. Therefore, a worm infection rate is influenced by the degree of homogeneity of a population of Bluetooth-enabled devices. While the likelihood of a vulnerability is higher in a heterogeneous population, a worm infection could cause a larger degree of damage in a homogeneous population.

Unfortunately, our traces do not reveal the types of the Bluetooth software stacks running on the discovered devices. However, we use the devices' manufacturers as a first-degree approximation of the degree of homogeneity of the Bluetooth stack implementations. While Bluetooth stacks running on different devices made by the same manufacturer are not necessarily identical, there is evidence that Bluetooth vulnerabilities do affect different versions and devices as long as they are made by the same manufacturer [13].

In Figure 2, we provide a breakdown of the Bluetooth-enabled devices based on their manufacturer. While we find a large number of manufacturers in each of our traces (11 in Pacific Mall, 13 in Eaton Centre, and 11 in the subway trace), Sony Ericsson and Nokia together account for half of all devices found. These results suggest that a software vulnerability present in the Sony and Nokia implementations of the Bluetooth protocol could affect more than half of all devices found in our traces. While Motorola is a popular manufacturer of many of the cell-phones available in the Toronto area, we found very few Motorola devices in our traces. Upon investigation, we found that most Motorola cell-phones are set to be non-discoverable by default.

5.4 Contact Durations are Long Enough for a Worm to Replicate

An infection occurs when a source device is in contact with a target long enough for the worm to replicate itself. Consequently, how fast a worm spreads is influenced by how long devices remain in contact. We can provide initial insight into how long devices remain in contact by examining the contacts' durations between our scanners and the devices encountered.

On the left, Figure 3 shows the distribution of the contacts' durations between our scanners and other devices. Many discovered devices responded to our pings only once; we marked these contacts as having a duration of 0 seconds. We find that a large fraction of contacts (52%) are short, lasting less than five seconds, while 30% of all contacts last more than 16 seconds.

On the right, Figure 3 shows the throughput and the failure rate of transmissions between two devices we controlled. We transfered a 256KB file between two devices placed apart at different

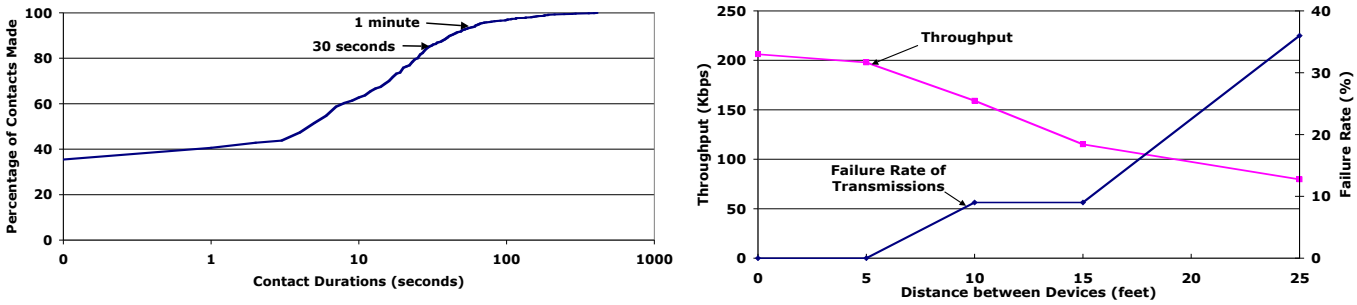

Figure 3: Characteristics of Bluetooth connections: The distribution of contact durations across all three traces is presented on the left. On the right, we conduct experiments of how Bluetooth throughput varies with distance inside the malls and the subway. We performed eleven trials for each inter-device instance per trace, and we report average values across the trials. As distance increases, Bluetooth transmissions start to fail.

distances, varying between 0 feet (i.e., next to each other) to 25 feet. For each distance, we performed eleven transfers while tracing inside the malls and the subway, and we present the average throughput.

As Figure 3 shows, the throughput declines almost linearly with the inter-device distance. While the lowest measured throughput was 7Kbps and occurred when the devices were 25 feet apart, the highest throughput was over 220Kbps. Across all throughput measurements performed, we find that the throughput between two devices is 185Kbps, on average. The failure rates increase with the distance between two devices. Combining all these results, half of contacts made while tracing in the malls and on the subway were long enough to transfer over 115KB, assuming an average throughput of 185Kbps. We believe that 115KB is sufficient for a worm to replicate its code; some of the most damaging Internet worms are less than 100KB in size [22].

5.5 Worms can Replicate between Devices Moving in Opposite Directions

Based on our trace, we found that many contacts are sufficiently long for two devices to exchange a worm. In this subsection, we examine whether an infection source is less "effective" when moving by "missing" all devices moving in the opposite direction. For this, we use controlled experiments to determine whether a worm can replicate between two devices moving in opposite directions at human walking speeds. We conservatively assume that two Bluetooth devices must establish a Bluetooth connection for a worm infection to occur. More harmful attacks that exploit vulnerabilities in the Bluetooth inquiry phase do not need a Bluetooth connection completely established to infect a victim device.

Figure 4 illustrates the average duration of transferring 500 bytes and 25KB between two devices when they move at a relative speed of 1 m/s and 2 m/s, respectively. A relative speed of 1 m/s corresponds to a scenario when one device is stationary and the other is moving at human walking speeds. A relative speed of 2 m/s corresponds to a scenario when both devices are moving in opposite directions. As Section 2 presented, there are three phases of a Bluetooth connection setup: the inquiry phase, setting up a link, and creating a socket. We chose to use an asynchronous link (ACL) in our experiments. With ACL, lost packets are retransmitted until their successfully acknowledged. In this way, we ensure that the data has been transferred reliably between devices. As Figure 4 shows, the Bluetooth connection spends most of the time (about 92%) in the inquiry and ACL-setup phases. Because little data is exchanged, the two setup phases dominate the time spent transferring the data. We deliberately kept the amount of data exchanged

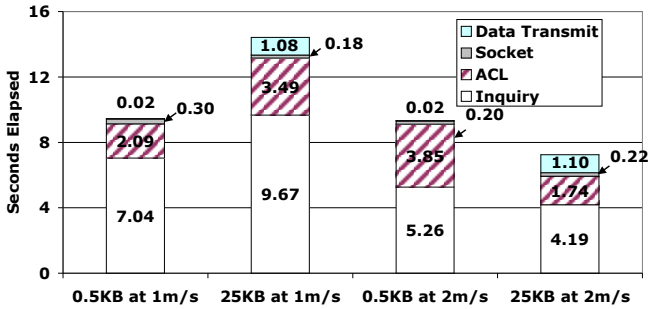

Figure 4: Bluetooth transmissions durations in a controlled experiment: We breakdown transmissions in four steps: the inquiry step, ACL step, socket creation step, and, finally, data transmission. Each of the bars represent the average of five experiments varying the amount of data transmitted and the speed of the mobile device.

low, to better match the scenario in which a hypothetical worm replicates its code. We also find that the average Bluetooth transfer is typically faster at 2 m/s than 1 m/s. The explanation for this counter-intuitive finding is that faster speeds put the devices within line-of-sight more quickly.

Our results illustrate that worms can infect at typical human walking speeds. The average Bluetooth connection's duration is anywhere between 2.2 to 15.9 seconds, enough for two humans walking at 1 m/s in opposite directions to exchange a copy of the worm.

5.6 Summary

This section examined whether the outbreak of a Bluetooth worm infection is viable today. In summary, we found that:

1. Discoverable Bluetooth-enabled devices are prevalent today.
2. The device population is relatively homogeneous.
3. Most devices remain within a scanner's Bluetooth range long enough for an infection to occur.
4. Walking cannot prevent a person's device from becoming infected.

6. SIMULATING BLUETOOTH WORM PROPAGATION

The previous section showed that the outbreak of a Bluetooth worm is viable today. This section uses simulation to explore the propagation dynamics of Bluetooth worms. This simulation captures several key factors of a worm infection, such as the number of initial devices infected (i.e., the number of "seeds"), the total size of the device population, the fraction of vulnerable devices, and the time of the day when infections occur.

Gathering a suitable trace to simulate Bluetooth worm infections is difficult. Such a trace requires tracking thousands of devices simultaneously while recording all interactions among them. To simulate how a worm spreads worldwide, a trace needs to capture the behavior of many devices at a global scale. Our trace presented in Section 5 is inadequate to simulate a Bluetooth worm infection. The goal of our trace was to capture all Bluetooth activity in an environment suitable for starting a worm outbreak. We instrumented our scanners to discover new devices as aggressively as possible. We did not capture any interactions among the devices discovered. As a result, our trace is unrepresentative of the behavior of typical vulnerable Bluetooth devices.

In the absence of an adequate trace, we build a preliminary model that approximates the behavior of a large device population from a trace of a small number of Bluetooth-enabled devices. For this, we use a trace of Bluetooth activity previously gathered by the Reality Mining project at MIT [11]. In this project, 100 students carried cell-phones instrumented to discover nearby Bluetooth devices for 18 months in 2004 and 2005. The scanning frequency was once every five minutes. We assume that these devices' interactions are diverse enough to represent the interactions within a device population.

6.1 Methodology

In all our simulations, we use a trace of Bluetooth activity spanning one month only. While the MIT project's data spans several months, we found that a one month trace contains sufficient events for our simulations. After examining the trace, we chose a 30 day period with average Bluetooth activity, starting on Thursday, October 7th, 2004 at 9:00am. Our trace consists of 51,316 events. Each event is a four-tuple: a sender, a receiver, discovery start time, and discovery end time. Overall, there are 80 unique senders and 3,833 unique receivers in our trace.

Our simulator takes four inputs: the device population size, the fraction of vulnerable devices, the number of infection "seeds" (i.e., the number of initially infected devices), and the input trace of Bluetooth events. On setup, the simulator inserts all input trace's events in an event queue ordered by the events' start times, and designates the "seeds" and the vulnerable devices. Once the setup is complete, the simulator dequeues each event, and checks whether the sender is already infected and whether the receiver is vulnerable. In this case, an infection occurs. If the node infected is not among the devices being traced, the input trace has no associated events in which the node acts as a sender. In this case, the simulator creates artificial events for the new node and it inserts them in the event queue.

We use a simple scheme to create artificial events for a new node. Each new node is mapped to an old sender from the input trace. The new node "inherits" all start times and end times of all events associated with the old sender. By not modifying the times associated with each event, the simulator preserves any time of the day effects present in the input trace. For each event, the simulator chooses a new receiver out of the entire device population. We use the following heuristic to assign receivers to a new device: devices appearing as receivers more frequently in the input trace are more likely to become the new node's receivers. Each device is selected as a new receiver with a probability directly proportional to the number of unique senders discovering this device in the input trace. The intuition behind our heuristic is that "popular" devices in the trace are likely to remain "popular". One effect of our heuristic is that worms spread more slowly because new devices are more likely to encounter popular "older" nodes.

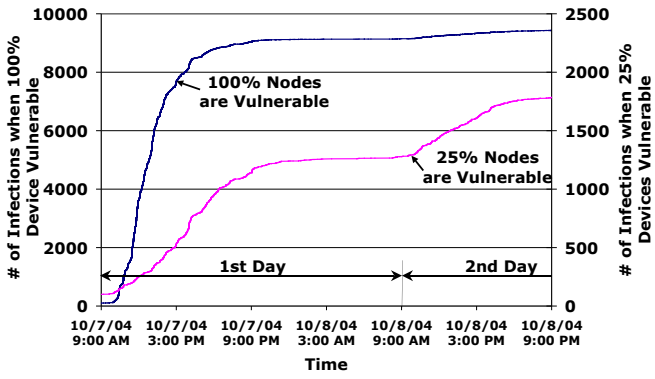

Figure 5: The propagation of a Bluetooth worm over time: This simulation uses a preferential attachment model in which newly created nodes are more likely to encounter popular nodes than unpopular ones. A popular node is one encountered by many others. This graph presents two simulations: one in which all 10K devices are vulnerable (y-axis on the left), and one in which only 25% of devices are vulnerable (y-axis on the right). We use 100 seed devices in both simulations.

Our simulation has several limitations. First, it does not capture the physical proximity and the geographical distribution of these devices. Second, it assumes that when any two devices are in contact, an infection occurs immediately. Third, the input trace is likely to be unrepresentative of how a Bluetooth worms spread around the world. We believe that our simulations match more closely the spread of a worm within one single community, such as a city. In consequence, all our simulations use a device population of 10,000 Bluetooth devices. We plan to address these limitations in future work.

Our model is different than traditional epidemic models, such as the SIR model [31]. First, traditional epidemic models are *homogeneous*, in the sense that an infected device is equally likely to infect any other susceptible device. Second, our model makes an attempt to preserve a non-uniform degree of popularity across the entire device population. Third, our model reproduces any temporal effects present in the data, such as time-of-the-day or day-of-the-week effects.

6.2 Bluetooth Worm Infection Dynamics

In this subsection, we examine how quickly a worm infects vulnerable nodes in a population of 10,000 devices. We conduct two experiments: one in which all devices are vulnerable, and one in which a vulnerability is discovered in a single manufacturer's Bluetooth implementation. In our experiments presented in Section 5, we found that a single manufacturer (e.g., Sony) can account for at least 25% of all Bluetooth devices today. Therefore, we chose to simulate a scenario when only 25% of devices are vulnerable.

Figure 5 shows the results of our simulations: one in which all 10,000 devices are vulnerable, and one in which only 2,500 devices (25%) are vulnerable. In both cases, the simulations start with 100

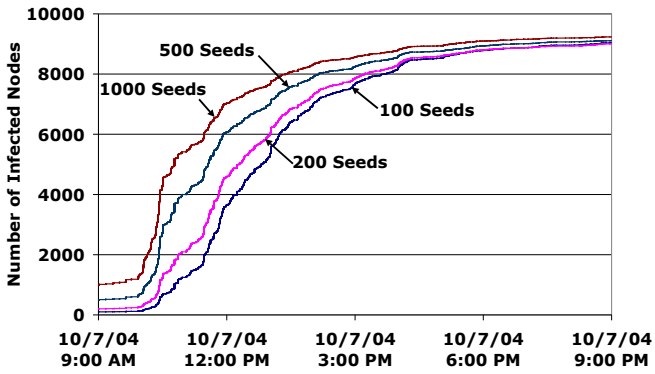

Figure 6: Varying the initial number of infection seeds: While more seeds make worms spread more quickly, the infection rates' speedups are relatively modest. We use 100 seed devices and 100% of the population is vulnerable.

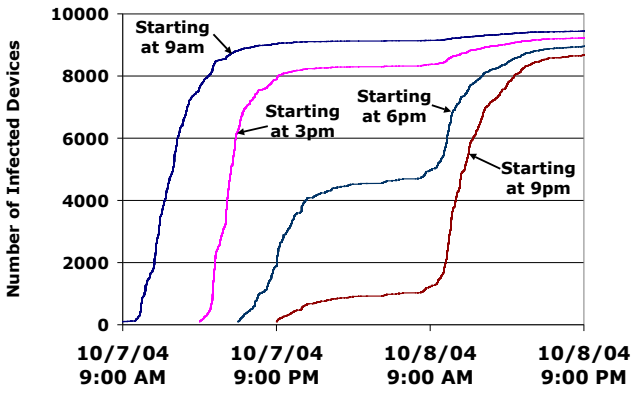

Figure 7: Varying the initial time of the outbreak: The time of the day affects Bluetooth worms' infection rates. Bluetooth worms spread more quickly during the day than over night. We use 100 seed devices and 100% of the population is vulnerable.

infected devices, i.e., "seeds". We find that a Bluetooth worm infects half of the devices in four hours when the entire device population is vulnerable. When only 25% of devices are vulnerable, half of them are infected in sixteen hours. In both cases, the worm propagates quickly: in 24 hours, the worm reaches a large fraction of vulnerable devices (90% and 51%, respectively.) We can also see that a worm "slows-down" during nighttime, but it then resumes a quick pace of infections the following day. These preliminary findings suggest that a Bluetooth worm can infect a significant number of devices in just a few days.

6.3 Increasing the Initial Number of Infection Seeds

In this subsection, we examine whether the number of initial infections affects how quickly a worm spreads. This can help us gain insight into whether an attacker needs a large number of initial victims before a worm starts to propagate rapidly.

Figure 6 shows the infection rate of a Bluetooth worm in a population of 10,000 devices when the number of seeds is set to 100, 200, 500, and 1,000 respectively (all devices are vulnerable). We find that more seeds make a worm spread more quickly, although the infection rate speedups are modest. This suggests that the number of initial seeds does not strongly affect the spread of the worm.

6.4 Varying the Initial Time of the Outbreak

In this subsection, we examine whether the initial time of outbreak affects how quickly a worm propagates. Figure 7 shows the infection rate of a Bluetooth worm when its outbreak occurs at 9am, 3pm, 6pm, and 9pm, respectively. As before, we simulate a population of 10,000 devices with 100 devices initially infected. All devices are vulnerable. We find that the worm discovers few vulnerable devices during nighttime. Therefore, a worm's infection rate is initially slow if the infection outbreak occurs during off-peak hours. Although not presented here, we also found that Bluetooth worms spread more slowly when their outbreaks occur on weekends and holidays.

6.5 Summary

This section used preliminary simulations to examine the propagation dynamics of Bluetooth worms in a population of 10,000 devices. We found that:

1. Bluetooth worms spread quickly, infecting a population of 10,000 devices over a few days only. If all devices are vulnerable, 90% of them are infected in 24 hours. If only 25% of them are vulnerable, the majority are infected in less than two days.

2. Bluetooth worms spread more quickly when more initial seeds are infected. However, the effects of the number of seeds on a worm's infection rate are modest.

3. Bluetooth worms spread more quickly during the day than over night.

These results have implications for the security of Bluetooth devices as a whole, and for the design of detection and prevention systems. In the next section, after we present our conclusions, we discuss these implications.

7. CONCLUSIONS AND DISCUSSION

This paper presented a preliminary investigation of worms in a Bluetooth environment. We showed that the Bluetooth protocol is complex and it is already facing a diverse set of known security attacks. Based on traces of Bluetooth activity, we found that starting a Bluetooth infection today is easy, once a vulnerability is discovered. Finally, we used trace-driven simulations to examine the propagation dynamics of Bluetooth worms. We found that Bluetooth worms can infect a large population relatively quickly, in just a few days.

Many of today's Bluetooth devices provide the user with the option to make them non-discoverable. Once a device is made non-discoverable, the device does not respond to any received Bluetooth inquiry messages. This feature suggests that an effective countermeasure to a Bluetooth worm infection is making the device non-discoverable or even turning the Bluetooth radio off. We find this solution unappealing: it will prevent devices from using Bluetooth for legitimate applications [21]. We believe that solutions that prevent worm infections while preserving Bluetooth functionality are preferable.

Although Bluetooth worms spread orders of magnitude more slowly than Internet worms, Bluetooth worms spread quickly enough that human-mediated counter-response solutions are likely to be difficult to implement in practice. Based on our simulations, such solutions must detect a worm's presence, analyze infected code, and create, test, and distribute a security patch in the span of

several days. Making a human-mediated response practical will be a challenging task.

Another implication of our data and simulations is that starting a Bluetooth worm outbreak is relatively easy once a vulnerability is found. An attacker can bring an infected device into a typical urban mall and discover many potential victims. The attacker does not need to devise a strategy about maximizing the number of infections: typically there is enough time to infect any device within the attacker's Bluetooth range. All these findings suggest that tracking where or how the infection started will be difficult.

The Bluetooth stacks of many of today's cell-phones are burned in read-only memory (ROM), making it difficult to patch their vulnerabilities. Nevertheless, if patching security vulnerabilities is viable, distributing the patches should be done during nighttime and week-ends when many cell-phones are inactive, minimizing the users' inconvenience. These off-peak time periods are sufficiently long that cell-phone providers could automatically upload, install, and test software updates. This entire process could be made user transparent.

Because Bluetooth worms spread quickly, a monitoring system placed in a high-traffic location could detect a worm infection early. A high-traffic location, such as an airport or train station, could also serve as an adequate deployment point for quarantine-based solutions: detecting a worm infection in an airport and isolating it could help prevent it from spreading globally.

8. ACKNOWLEDGMENTS

We wish to thank Alec Wolman, Krishna Gummadi, and the anonymous reviewers for their comments and feedback. We gratefully acknowledge the use of Bluetooth data from Nathan Eagle at MIT and the CRAWDAD [9] archive at Dartmouth College. We also acknowledge Anna Popivanova who first argued that computer worms will spread over cell-phones in the future. This research has been supported by NSERC grant # 205834-480534.

9. REFERENCES

[1] F. Armknecht. A Linearization Attack on the Bluetooth Key Stream Generator, 2002. Cryptology ePrint Archive, Report 2002/191.

[2] Bluetooth. Specification of the Bluetooth System, 2006. http://www.bluetooth.org/foundry/adopters/document/Core_v2.0_EDR/en/1/Core_v2.0_EDR.zip.

[3] Bluetooth.com. The Official Bluetooth Wireless Info Site, 2006. http://www.bluetooth.com.

[4] BlueZ. BlueZ – Official Linux Bluetooth Protocol Stack, 2006. http://www.bluez.org.

[5] T. Bunker. Serious Flaws in Bluetooth Security Lead to Disclosure of Personal Data, 2006. http://www.thebunker.net/security/bluetooth.htm.

[6] R. G. Cole. Initial Studies on Worm Propagation in MANETS for Future Army Combat Systems, 2004. http://stinet.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA431999.

[7] R. G. Cole, N. Phamdo, M. A. Rajab, and A. Terzis. Requirements of Worm Mitigation Technologies in MANETS. In *Principles of Advanced and Distribution Simulation*, 2005.

[8] ComputerWorld. Cabir Worm Wriggles into U.S. Mobile Phones, 2005. {http://www.computerworld.com/securitytopics/security/virus/story/0,10801,99935,00.html}.

[9] CRAWDAD. Crawdad: A Community Resource for Archiving Wireless Data at Dartmouth, 2006. http://crawdad.cs.dartmouth.edu/l.

[10] D. Dagon, T. Martin, and T. Starner. Mobile Phones as Computing Devices: The Viruses are Coming! *IEEE Pervasive Computing*, 3(4):11–15, 2004.

[11] N. Eagle and A. Pentland. Reality Mining: Sensing Complex Social Systems. *Journal of Personal and Ubiquitous Computing*, June 2005.

[12] S. R. Fluhrer. Improved Key Recovery of Level 1 of the Bluetooth Encryption System, 2002. Cryptology ePrint Archive, Report 2002/068.

[13] M. Herfurt. Bluetsnarfing @ CeBIT 2004 – Detecting and Attacking Bluetooth-enabled Cellphones at the Hanover Fairground, 2004. http://trifinite.org/Downloads/BlueSnarf_CeBIT2004.pdf.

[14] M. Hermelin and K. Nyberg. Correlation Properties of the Bluetooth Combiner Generator. In *Information Security and Cryptology*, pages 17–29, 1999.

[15] B. Hoh and M. Gruteser. Computer Ecology: Responding to Mobile Worms with Location-Based Quarantine Boundaries. In *International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks*, 2006.

[16] InfoSyncWorld. First Symbian OS Virus to Replicate over MMS Appears, 2005. http://www.infosyncworld.com/news/n/5835.html.

[17] M. Jakobsson and S. Wetzel. Security Weaknesses in Bluetooth. *CT-RSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology*, pages 176–191, 2001. LNCS 2020.

[18] A. Laurie, M. Holtmann, and M. Herfurt. Bluetooth Hacking, 2004. http://www.ccc.de/congress/2004/fahrplan/event/66.en.html.

[19] O. Levy and A. Wool. A Uniform Framework for Cryptanalysis of the Bluetooth E0 cipher, 2005. Cryptology ePrint Archive, Report 2005/107.

[20] Y. Lu and S. Vaudenay. Faster Correlation Attack on Bluetooth Keystream Generator E0. In *Advances in Cryptology (CRYPTO)*, Santa Barbara, CA, 2004.

[21] Mobileinfo.com. Bluetooth Technology – What are the Applications?, 2006. http://www.mobileinfo.com/Bluetooth/applic.htm.

[22] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The Spread of the Sapphire/Slammer Worm. *Technical Report CAIDA, ICSI, Sillicon Defense, UC Berkeley EECS and UC San Diego*, January 2003.

[23] D. Moore, C. Shannon, and J. Brown. Code-red: a case study on the spread and victims of an internet worm,. In *Proceedins of the 2002 Internet Measurement Workshop*, November 2002.

[24] E. O'Neill, T. Kindberg, A. F. gen Schieck, T. Jones, A. Penn, and D. S. Fraser. Instrumenting the city: developing methods for observing and understanding the digital cityscape. In *Proc. of the 8th International Conference on Ubiquitous Computing (UBICOMP)*, 2006.

[25] Palm. Bluetooth technology: what is it, how does it work, and what can I do with it?, 2006. http://kb.palmone.com/SRVS/CGI-BIN/WEBCGI.EXE?New,Kb=PalmSupportKB,ts=Palm_External2001,case=obj(20821).

[26] PCWorld. What's Cooking? Bluetooth Hits the Kitchen, 2002. http://www.pcworld.com/news/article/0,aid,95223,00.asp.

[27] T. Register. Bluetooth to Outship Wi-Fi Five to One, 2003. http://www.theregister.co.uk/2003/06/18/bluetooth_to_outship_wifi_five/.

[28] Y. Shaked and A. Wool. Cracking the Bluetooth PIN. In *Proceedings of 3rd USENIX/ACM Conference of Mobile Systems, Applications and Services (MOBISYS)*, June 2005.

[29] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proc. of 2002 USENIX Security Symposium*, 2002.

[30] O. Whitehouse. Bluetooth: Red Fang, Blue Fang, 2004. http://www.cansecwest.com/csw04/csw04-Whitehouse.pdf.

[31] Wikipedia. Compartmental models in epidemiology, 2006. http://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology.