Studying Bluetooth Malware Propagation: the BlueBag Project

Luca Carettoni* Claudio Merloni† Stefano Zanero‡

November 15, 2006

Abstract

The real danger posed by Bluetooth malware is still debated. Is it just "yet another" form of viral code, which creates no new issues, and with a relatively low chance of creating virtual epidemics? If current Bluetooth worms pose relatively little danger when compared to Internet scanning worms, in this article we envision targeted attacks through Bluetooth malware, and create proof of concept codes and devices demonstrating their feasibility.

We demonstrate how it is possible to build a distributed network of bots, spreading via Bluetooth, which can propagate through Bluetooth flaws and risky user behavior, target specific devices, and exploit them to log keystrokes, steal data, record audio data, take pictures, and ultimately send the collected data back to the attacker, either through the agents network or directly. We show the different elements that compose the whole project and we also give estimates, through real data and simple simulations and models, of the effectiveness of such an attack.

1 Introduction and motivations

Mobile computing is quickly gaining ground in our daily experience, as a pervasive business and life-enabling technology. For this reason, it is very important to understand the potential risks linked with all types of wireless devices and communication protocols.

Bluetooth, thanks to its characteristics, will become the pervasive technology to support wireless communication in various contexts of everyday life. At present, the greatest level of diffusion is witnessed in so-called smart phones. These devices, on top of offering all the functions of a cutting-edge telephone, integrate the functions of an advanced handheld computer, managed by an operating system, such as Symbian or Microsoft Windows Mobile.

Smart phones allow sending and receiving SMSs, MMSs and e-mail, listening to MP3 files, watching videos, surfing the Internet, playing games, managing an agenda, synchronizing phone data with PC data and much more. Albeit still constituting a niche market, they observed a growth rate of 100% per year for

*Secure Network Srl, Agrate Brianza, Italy, l.carettoni@securenetwork.it

†Secure Network Srl, Agrate Brianza, Italy, c.merloni@securenetwork.it

‡Contact author, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano Italy, zanero@elet.polimi.it

the past 5 years, and according to ABI Research, a market research company, this year they will reach 15% of the global cellular phone market, equivalent to 123 million units sold, thanks to the growing request for applications such as mobile e-mail (which according to Gartner, this year will be used by 20 million people), to decreasing prices and to the broader choice of models.

Smart phones, thanks to the advanced functions of communication and productivity they provide, are now very similar to personal computers: because of this, they are at the same time more vulnerable, more useful and more attractive for a potential attack than previous types of portable phones. The increased vulnerability is due to the presence of a system of evolved connectivity applications that expose the telephone (and the data it contains) to all the risks that arise from activities such as sending e-mail, transferring data through MMS, Internet and Bluetooth communication, the use and exchange of memory cards.

In this article, we focus on the new risks created by the widespread presence of Bluetooth-enabled devices that have potentially sensitive data and vulnerability-prone software. In particular, we study how this mix of technologies could become a potential vehicles for the propagation of malware, specially crafted to extract information from smartphones.

Cellular phone viruses spread at present have fortunately not caused significant damage to other users, except for the obvious inconveniences due to telephone malfunctioning. This created the myth that Bluetooth malware is just "yet another" form of viral code, which does not pose real new security issues, and with a relatively low chance of creating damages. In this work, we aim to demonstrate scenarios through which future Bluetooth malware could attack our devices.

We will show proof of concept codes and devices that demonstrate demonstrate the conjunct exploitation of Bluetooth device weaknesses and user behavior to build a distributed network of agents, spread via Bluetooth, which can seek given targets and, once reached, exploit the devices capabilities to log keystrokes, steal data, record audio data, take pictures, and ultimately send the collected data back to the attacker.. We will also show the different elements that compose the whole project and give estimates, through real data and simple simulations and models, of the effectiveness of such an attack.

2 An overview of Bluetooth technology and security issues

Bluetooth is a word that is now commonly used. The literal meaning supposedly refers to the Viking Emperor Harald (Bltand in Scandinavian), who lived at the beginning of the 10th century and united the kingdoms of Denmark, Norway and Sweden. The objective of the Bluetooth protocol is in fact to unify different wireless data transmission technology among mobile and static electronic devices such as PCs, cellular phones, notebooks, palm pilots, DVDs, MP3 devices, TV, Hi-Fi, cash registers, POS terminals and even household appliances such as refrigerators and washing machines.

Basically, Bluetooth is an alternative to traditional infrared communication standards (e.g. IrDA), and is based on a short-wave radio technology, which is able to transmit data across physical obstacles such as walls or other objects [12].

Bluetooth devices use the 2.4 GHz frequency range (the same range used by Wifi IEEE 802.11 technology), letting devices covered by the signal communicate among each other. The exact frequency spectrum used varies from country to country due to national regulations.

An important improvement over IrDA is the lack of the requirement of line of sight among devices, and the increased range of connectivity. When an individual connects different Bluetooth devices together, he creates around himself a so called PAN (Personal Area Network), a small ad hoc network with the possibility to exchange data and information as it usually occurs within a regular company LAN (Local Area Network). This is one of the key reasons why Bluetooth can be used, for instance, as a transport for automatically spreading malware, while this is not the case with IrDA, since the latter required a proper alignment of transmitting and receiving device which effectively avoided "casual" or unwanted interaction.

Bluetooth technology is characterized by a low power (from 1 to 100 mW, a thousand times less than the transfer power of a GSM cellular phone) and a communication speed of around 1 Mbps. With regards to power, Bluetooth devices can be grouped in classes, each corresponding to a different reach:

- Class 1: able to communicate with Bluetooth devices in a 100 m range
- Class 2: able to communicate with Bluetooth devices up to a 10 m range
- Class 3: able to communicate with Bluetooth devices within a 1 m range

Currently, most common devices belong to classes 2 and 3: for instance notebooks and cellular phones normally use Class 2 peripherals. Towards the end of 2004, a new implementation of the Bluetooth technology (version 2.0) was released, allowing for transfer speeds of up to 2 and 3 Mbps, as well as lower energy consumption. The new protocol is backward-compatible.

The Bluetooth standard incorporates very robust security mechanisms [5] that can be used to create very secure architecture. A series of theoretical glitches and possible attacks were discovered in the core specifications of Bluetooth [9, 6]. The most serious of these (described and implemented in [16]) can lead to a compromise of the cryptographic algorithm protecting communication through sniffing, but this is less than practical since the attacker needs to be present to the pairing of devices, and to be able to sniff communications among them. This is more difficult than it would appear, since Bluetooth divides the 2.4 GHz spectrum range into 79 channels, through which devices hop with a pseudorandom hopping sequence which is different from PAN to PAN. This is done both to avoid interferences among different PANs and for security enhancement. In fact, this inhibits common Bluetooth hardware from sniffing communications in a PAN they don't take part in. Alternate implementations of Bluetooth with more secure encryption algorithms have also been proposed [2].

Even if Bluetooth is theoretically quite robust, since late 2003, a number of security issues in various specific implementations of the standard stack surfaced. Among the existing attacks, we can quote some significant examples drawn from [1]:

BlueSnarf: This type of attack uses the OBEX Push service (the type of service commonly used to exchange files such as business cards). BlueSnarf allows to access the phone book and the agenda of vulnerable devices without authentication. A recently upgraded version of this attack gives the attacker full read/write access.

Bluejacking: By sniffing the IDs that devices exchange during devices association, short deceitful text messages can be transmitted. A user could then be tricked into using his own access code and authorize an aggressor to access a phone book, agenda or file residing on the device

BlueBug: This vulnerability allows to access the AT Commands of the cellular phone, which allow an aggressor to use the phone services, placing outgoing phone calls, sending, receiving or deleting SMS messages, diverting calls, and so on.

BlueBump: This attack takes advantage of a weakness in the handling of Bluetooth link keys, giving devices that are not authorized anymore the ability to access services as if still paired. This can lead to data theft, or to the abuse of WAP and GPRS services.

BlueSmack: This is a Denial of Service attack that knocks out some types of devices, and can be conducted using standard tools

HeloMoto: This is a combination of BlueSnarf and BlueBug. The name comes from the fact that it was originally discovered on Motorola phones

BlueDump: This attack causes a Bluetooth device to 'dump' its stored link key, creating an opportunity for key-exchange sniffing, or for another pairing to take place with the attacker's device of choice

CarWhisperer: CarWhisperer abuses the default configuration of many hands-free and headset devices, which come with a fixed PIN for pairing and transmission

BlueChop: this denial of service can disrupt any established Bluetooth piconet by means of a device that is not participating it, if the master of the piconet supports multiple connections

These flaws demonstrate how, in many cases, it is possible to steal information from mobile devices, controlling them from a distance, making calls, sending messages, or even connecting to the Internet. This type of problems is traditionally handled, in computer systems, with the release and application of patches. However, this approach does not extend to GSM handsets, since in most cases a firmware update can be performed only at service points and shops, not by the customers themselves: therefore many vulnerable phones and firmwares keep going around even long after a vulnerability is discovered.

Some of these attacks are implemented in "Bloover", a proof-of-concept application developed and released by Martin Herfurt, which runs on Symbian cellphones. This counters the idea that an attacker would need a laptop in order to execute these attacks, therefore making themselves visible. Some scripts and tools have also been ported to the Nokia 770 Internet Tablet. Most of these attacks can also be performed at a distance using long range antennas and modified Bluetooth dongles: a Bluetooth Class 2 device was reportedly able to perform a BlueSnarf attack at an astounding distance of 1.08 miles.

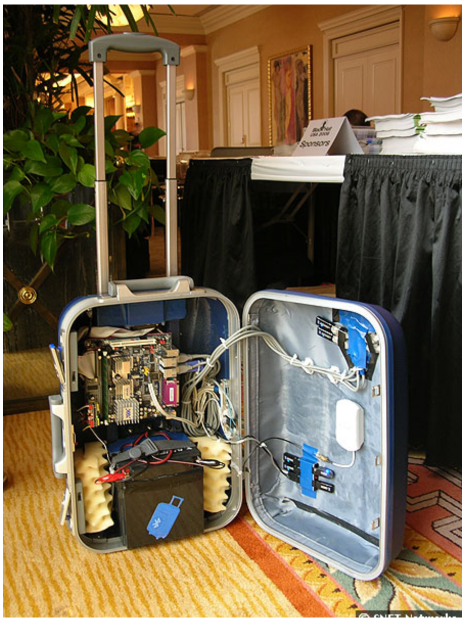


Figure 1: The BlueBag trolley (Photo courtesy of Cnet news.com)

3 Creating the BlueBag: a covert attack and scanning device

To carry out a survey, or an attack, without being noticeable, we needed to create a covert attack and scanning device, which we later came to call the "BlueBag" (see Figure 1).

We needed a Linux based system embedded with many Bluetooth dongles to process many discovered devices in parallel, as well as an omnidirectional antenna to improve the range and cover a wide area; we needed a hidden tool, but also an instrument which could be easily carried around and still have a long battery life.

To fulfill these requirements, the BlueBag was created by modifying a standard blue trolley with the insertion of a Mini-ITX system (see Figure 2), using the following off-the-shelf components:

VIA EPIA Mini-ITX motherboard: an integrated motherboard with a 600 Mhz Processor on board, one of the most powerful fanless motherboards available over-the-shelf (model PD6000E). Being fanless it reduces power consumption.

256MB of RAM: in a DDR400 DIMM module.

EPIA MII PCI Backplate: which extends the USB connections available on board from 2 to 6.

iPod 20 GB, 1.8 inch HD: a small size hard drive that can resist up to 3 G acceleration.

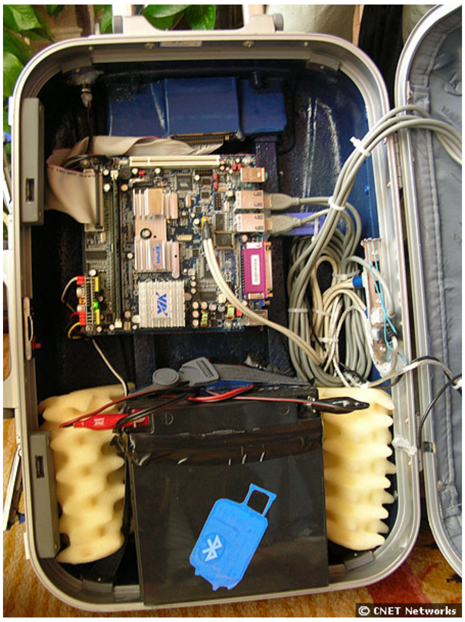


Figure 2: Motherboard (top) and battery (bottom) of the BlueBag (Photo courtesy of Cnet news.com)

8 Class-1 Bluetooth dongles: with a Broadcom chipset. Some of them are connected to a powered 4-port USB hub.

1 modified Class-1 Bluetooth dongle: a Linksys dongle (Cambridge Silicon Radio chipset), modified with a Netgear omnidirectional antenna with 5dBi gain.

picoPSU, DC-DC converter: the smallest snap-in 12V ATX power supply, it generates up to 120 Watts of power at over 96% efficiency.

12V-26Ah lead acid battery: provides the power needed for long sessions of surveying. With the power consumption of the BlueBag equipment, it provides more than 8 hours of power autonomy.

The total cost to build a device such as the BlueBag is about $750, demonstrating that Bluetooth attack devices can be extremely economical and easy to create.

The BlueBag runs on GNU/Linux OS (specifically, we use the Gentoo distribution for its outstanding customizability and performance), on the top of which we created a software infrastructure in Python which makes it easy to devise, control and perform survey sessions. The software is completely multi-threaded and the available dongles can be used concurrently to perform different tasks. We implemented a simple but useful dongle management and allocation, to dynamically learn about available resources and lock them when needed. Dongles can be reserved to be able to run applications that need a fixed interface (e.g.. pan daemon in background). The software is quite modular an designed with the typical producer/consumer pattern. Producers put found devices in a queue, using the standard utilities that come with BlueZ (the official

Linux Bluetooth stack) in order to collect information about devices, as well as customized versions of well known Bluetooth information gathering techniques such as blueprinting (a method to remotely identify Bluetooth-enabled devices, similar to OS fingerprinting). A distinct thread manage the queue and assign tasks to different consumers.

The BlueBag software suite allows for monitoring and controlling the execution of the test from a palmtop or smartphone, through the use of a Web interface which runs on top of a TCP/IP over Bluetooth connection. In this way, there is no need to open the BlueBag case while walking around. It is worthy of note that in no case we were stopped or suspected of doing something unusual.

4 Survey results: a discomforting landscape

In our surveys, we initially focused on the identification of the number of active Bluetooth devices in discoverable (or visible) mode. This is in fact the condition of real potential risks: it has been demonstrated that through a brute-force attack it is possible to find devices with active Bluetooth technology in non-discoverable mode. However, this is unfeasible in a generic context, given the enormous time expenditure it would entail. An attack with this method is possible only if a specific device is targeted and even in this case it is first necessary to identify the brand and model of the device, in order to be able to prune the address space.

Therefore, keeping a phone in non-discoverable mode gives a certain protection against targeted attacks, and in general keeps the device also safe from worm infections that use Bluetooth technology to replicate, since the research of victim devices occurs through a simple scanning of devices in the area. For this reason, our test was focused exclusively on the detection of devices in discoverable mode, which are the only ones actually in a condition of potential risk of attack from Bluetooth malware.

We decided to conduct surveying in several high-transit locations surrounding Milan:

- Milan's Fair, during InfoSecurity 2006 trade show
- Orio Center Shopping Mall
- MM2 Cadorna Metro Station
- Assago MilanoFiori Office District
- Milan Central Station
- Milano Malpensa Airport
- Politecnico di Milano Technical University, Leonardo Branch

We chose different types of venues in order to evaluate if and how the prevalence of potentially vulnerable targets varied in different contexts populated by different people: At the Central Station, for instance, there is a very heterogeneous user base (and a dense crowd: the station serves 270000 passengers on an average business day); at the Orio Center on a Saturday there are many young

Location	Date	Duration (hh:mm)	Unique Devs	Devs Rate
InfoSecurity 2006	02/08-10/06	4:42	149	0.53
Orio Center Shopping Mall	03/01-11/06	6:45	377	0.93
MM2 Metro Station	03/09/06	0:39	56	1.44
Assago Office District	03/09/06	2:27	236	1.60
Milan Central Station	03/09/06	1:12	185	2.57
Milano Malpensa Airport	03/13/06	4:25	321	1.21
Politecnico di Milano T.U.	03/14/06	2:48	81	0.48
		22:58	1405	

Table 1: Summary of surveying results

people and families, subjects that could potentially be unaware of dangers linked with new technologies, as opposed to visitors and exhibitors at the Infosecurity trade show (about 2000 professional operators a day).

We performed multiple sessions, on different days, for a total of 24 hours of scanning dispersed over 7 days. Results are shown in Table 1, where "Unique Devs" denotes the number of unique devices in discoverable mode which were found during the specific session, and "Devs Rate" the average number of unique device discovered per minute.

This data shows the capillary diffusion of Bluetooth technology in everyday life and also highlights the huge number of potentially vulnerable devices: this technology, at first sight, seems to be an integrating part of everyone's life, not only for professional activities but also for personal use. It can also immediately be noted that there are no significant differences in terms of risk awareness among the Central Station, the Milan Malpensa Airport (populated by a heterogeneous public) and the Assago Office District, where most users use these devices for work purposes. The situation was significantly better - indicating a greater awareness of users - at Infosecurity and at the University.

On 1405 unique devices detected, a further analysis was made to broadly categorize the devices: cellular and smart phones (1312), PCs/notebooks (39), palm pilots (21), GPS navigators (15), printers (5) and other various devices (13). In a similar experiment carried out in parallel and independently by F-Secure during CeBIT 2006 (the ICT trade show in Hannover), a regular laptop device capable of identifying active Bluetooth devices in a 100 mt range found, in a week of scanning, over 12.500 devices with discoverable Bluetooth mode. As of our knowledge, they made no attempt to break down the data any further, or to explore the offered services.

After that, we tried to analyze the type of services offered by the devices, and in particular those interested by the diffusion of worms. As shown in Table 2, the OBEX Push service was active and in range for enough time to allow

Service Type	Number of devices
OBEX Object Push, OBEX File Transfer	313
HeadsetHands-free Audio Gateway	303
Dial-up Networking	292

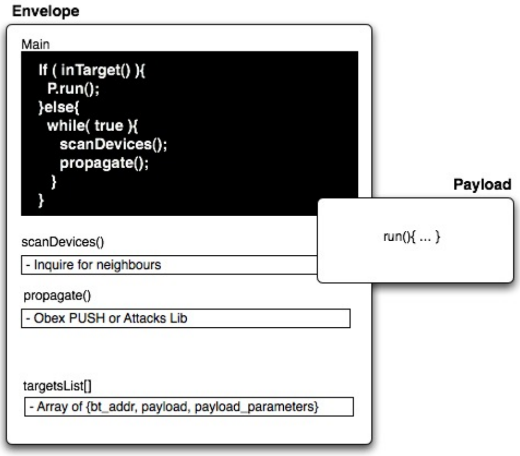
Table 2: Services offered by the devices

scanning on 313 devices; this is normally used for the transfer of information (business cards for instance) or files and application - including worms. Actually, it is very likely that most, if not all cellphones have the OBEX Push service activated. Since there are 1312 phones among the devices, the result could seem strange at first sight. The explanation is simple: among all those devices, 313 are the ones which stayed in the BlueBag's range long enough to allow the push service to be correctly polled.

Another important information that is possible to extrapolate from our survey is the "visibility time" of the devices, i.e. the average time in which a device remains in the range of a potential attacker, in others words the time in which an aggressor could exploit the device. This time depends substantially on the different activity patterns of people in different contexts: for instance, at the Orio Center Shopping Mall the average time was 12.3 seconds, at the Politecnico di Milano T.U. 10.1, in the Milano Malpensa Airport the time was 23.1 seconds. Of course, in some cases this time depends also on the activity pattern carried out by the hypothetical aggressor: for instance, at the Politecnico we deliberately avoided to stay in a single classroom for a long time, but an aggressor who is interested in a specific target may very well do so, or follow the target in an airport up to the gate (where most people settle down while waiting for boarding), and thus extend this time. The "average visibility times" we estimated are therefore interesting for casual contacts, such as the one

It is important to point out that some models of cellular phones are launched on the market with a configuration that, if the Bluetooth connection is activated, sets the discoverable mode as a default, requiring the user to manually modify the setting to the secure, non-discoverable mode. Other devices must instead be manually brought to discoverable mode, and are automatically reset to non-discoverable after a short period of time. Our survey showed this to be effective: just a handful of the detected device models were of this last type, surely out of proportion with the respective market shares. Since keeping devices in non-discoverable mode does not forbid communication among paired devices, but just occasional push communications among non-paired devices, keeping a phone in non discoverable mode should not entail a heavy usability burden.

After the first survey, to investigate the effectiveness of the propagation of Bluetooth malware we realized that we needed an estimation of the success rate of social engineering techniques used by common Bluetooth worms. Since most of the existing worms rely on the user accepting a file in order to propagate, we need an estimate of the ratio of users that will accept a file transfer of an unknown file from an unknown source. To obtain this data, we developed an "OBEX Pusher", an add-on to our normal survey scripts, which searches for all discoverable Bluetooth devices with OBEX Push support enabled, and sends them a file. Using this tool (and transmitting an innocuous image file) we



```
If ( inTarget() ){
  P.run();
}else{
  while( true ){
    scanDevices();
    propagate();
  }
}
```

Figure 3: Pseudocode of the Bluetooth worm with dynamic payloads for targeted attacks

could estimate that, with slight variations in the different locations analyzed, an astounding 7.5% of individuals carelessly accept unknown file transfers from unknown sources, and are thus be highly vulnerable to social engineering attacks.

5 Bluetooth-enabled Malware Networks

Our experiment shows that just a few people today are aware of the risks that can be incurred by using apparently innocuous devices. Moreover, as we already stated, smartphones and connected palmtops often are daily work tools for people with a medium/high level of responsibility within their companies. This implies that these devices may hold particularly interesting information that potential aggressors could be interested in, e.g. for industrial espionage.

All the elements are thus in place for creating a huge risk for companies and individuals: in the future, we can almost certainly foresee an increase of attacks, of viral nature or not, aiming not only to make a mobile device unusable, or to connect it to payable numbers in order to generate quickly illegal gains for the authors, but also targeted to the information it contains.

Since the reachability of the target device is often thought as a form of protection, we created (as a proof of concept) a network of viral agents that can spread through mobile devices looking for a target, zeroing in on it, and then reporting back information to the attacker.

Since such agents are targeted to a specific environment and person, makes it interesting to study the use of dynamic payloads that can vary depending on the type of infected device. We designed a proof-of-concept worm infrastructure which uses an envelope-payload mechanism, as shown in Figure 3.

The *envelope* component is a piece of software able to scan for Bluetooth devices and to propagate to found devices; it has a list of targets to propagate to, and a set of payloads that it can "deploy" on the targets. The *payload* components instead can be any type of malicious code that we want to execute on victim devices - within the limits of mobile phones operating systems. Examples of payloads could be, for instance, keyloggers, audio recorders, sniffers, etc. A similar design pattern (in a very different context) is found in the Metasploit framework's "Meterpreter" infrastructure [14].

Such payloads can also use the high connectivity of Bluetooth-enabled devices to transmit back to the attacker the information they have harvested (much in the same way as common PC-based spyware does), for instance using the Internet e-mail service, or a sequence of MMS. In this way there is no need for the attacked device to be in range with the attacker for sending home the retrieved data. A scenario which could take place is therefore the following: an attacker could infect (during the commute, for example) a number of devices belonging to the employees of an organization, and thereafter just wait for one of these devices to be able to infect or attack the device of their CEO. In other words, the attacker would create a botnet of Bluetooth enabled zombie machines, ready to do his bidding.

One of the barriers to the propagation of mobile malware has historically been the difference among various operating systems and hardware platforms. This is becoming increasingly less difficult due to the spreading of Java 2 Micro Edition (J2ME), which makes it easier for software authors (and correspondingly for malware authors) to create cross-platform software for mobiles. Our proof of concept is successfully implemented in Java and runs on any cellphone which is compatible with MIDP 2.0 and has JSR-82 activated (the Java Bluetooth API).

Features that would make this worm really dangerous (and which we therefore did not implement) are ways to auto-execute with as little interaction as possible with the device user. On Symbian phones, for instance, system files can be overwritten due to various structural flaws in access control. Otherwise, implementation flaws and bugs that allow for command execution (such as the ones we described before) could be used to help this worm propagate.

6 Simulation results

In order to correctly evaluate the threat posed by this attack vector in the scenario we developed, we developed a model and a simulation to understand the effectiveness of the propagation of Bluetooth malware. Due to space limitations, we refer the reader to [19, 4] for a discussion of the problems related with modeling the propagation of computer viruses. An excellent analysis of mathematics for infectious diseases in the biological world is instead available in [7].

Propagation models, however, evolve naturally, following the changes in the propagation vectors of viruses. The earliest models were targeted at virus propagation through the infection of host executables [10]. Most biological epidemiological models share two assumptions: they are *homogeneous*, i.e. an infected individual is equally likely to infect any other individual; and they are *symmetric*, which means that there is no privileged direction of transmission of the virus. The former makes these models inappropriate for illnesses that require a non-casual contact for transmission, as well as being inappropriate to describe the early stages of propagation of an epidemic which are strongly location-dependent; the latter constitutes a problem, for instance, in the case of sexually-transmitted diseases. In [10] these shortcomings are addressed by transferring a biological model onto a directed random graph in order to approximate the chain of software distribution and the way it worked in the early days of the personal computing revolution.

Among other results, the authors show that the more sparse[1] a graph is, the slower is the spread of an infection on it; and the higher is the probability that an epidemic condition does not occur at all.

The introduction of the Internet changed the malware landscape, and made such models unrealistic. The first effect was the appearance of mass-mailing worms, which demonstrated that tricking users into executing the worm code attached to an e-mail, or exploiting a vulnerability in a common e-mail client to automatically launch it, is a successful way to propagate viral code. One of the best models for such a propagation is described in [21], where the e-mail service is modeled as an undirected graph of relationships between people. The problem here lies in how to model the behavior of the user [20], if the worm does not automatically exploit a vulnerability, and in how to build the relationship graph.

Self-propagating worms, scanning for vulnerabilities, made their first appearance in [17], and in recent years changed the landscape of the threats once more. They can be modeled through the Random Constant Spread (RCS) model [18], developed using empirical data derived from the outbreak of the *Code Red* worm, a typical random scanning worm. This model uses an extremely rough approximation, ignoring the effect of immunization and recovery. It implicitly assumes that the worm will peak before a remedy begins to be deployed. Additionally, it models the Internet as an undirected, completely connected graph. This is far from being true [3], but still the model macroscopically behaves well. UDP-based worms require corrections in order to account for bandwidth restrictions and bottleneck Internet links [15].

The propagation of a Bluetooth virus can take place in several different ways. The most common, until now, is through simple *social engineering*. The worm sends messages with copies of himself to any device which comes into range through an OBEX push connection. The receiver, finding an "attractive" message on the cellular phone with the invitation to download and install an unknown program, often has no clue that this can pose a danger. For instance, Cabir, one of the first cellular phone worms, and the first case of malware able to replicate itself only through Bluetooth, used this technique.

MMS messages are another potential medium of propagation, e.g. the worm Commwarrior propagated also through MMS (in fact, it spread from 8 A.M. to midnight using Bluetooth connections, and from midnight to 7 A.M. through MMS messages). A final method of propagation, since most smartphones can use e-mail and potentially offer TCP/IP services, could be fairly similar to mail based or TCP based worms, such as the ones we usually witness on the Internet. This type of method has not been really used until now.

A variety of phone malware has already been found (by the end of May 2006, F-Secure research laboratories classified over 200 virus specimen) [11]. Of these, the largest majority of the ones commonly found in the wild propagate by relying only on the Bluetooth technology. In fact, our own experiments showed that this transmission method alone can reach 7.5% of a mixed population of

[1] In a sparse graph, each node has a small, constant average degree; on the contrary, in a local graph, the probability of having a vertex between nodes B and C is significantly higher if both have a vertex connected to the same node A.

targets. Therefore, we decided to simulate the propagation of viral code which uses Bluetooth as its vector.

This is not an easy task, actually. On one hand, we are interested in following the early stages of the propagation of a worm, since we want to evaluate its effectiveness as a targeted attack tool, not as a global infection. Therefore, the assumptions of homogeneity and non-locality cannot hold. On the other hand, we are simulating the interactions of highly mobile devices, which interact occasionally. So we are effectively simulating a highly sparse graph of relations which change dramatically over time.

In order to simulate the transient geographical relationships caused by the movement of people in physical places, we made use of interesting earlier results from the ad-hoc networks research community [13]. The tool described there (*CMMTool*) generates realistic traces of movement for people and their respective devices. We developed a small simulator which takes such feed as an input, and reproduces the behavior of a Bluetooth worm which is propagating across them. Named *BlueSim*, the resulting tool can replicate, under various hypotheses, the behavior of a real worm propagation, taking into account the visibility time of the devices, the inquire time needed, the data transfer rate and so on. We omitted to analyze layer-1 radio aspects such as collisions and interference problems, which could potentially occur in crowded places with many devices. In order to do so, a network simulator such as NS could be used [8].

In order to evaluate the effectiveness of propagation of a targeted worm through a population, we recreated different specific contexts with some fixed parameters drawn from real environment characteristics and data collected during our survey. In particular we simulated a shopping mall - which is a simplified version of the Orio Center mall which we visited - with 250×100 meters of surface and 78 shops. We considered a population of 184 discoverable devices (7.5% of which susceptible to infection), with a Bluetooth transmission range of 15 meters that is reasonable for mobile phones or PDAs. We conservatively estimated a 0.3 Mbps bandwidth link, and a huge 42 Kb worm, which is the effective size of the "envelope and payload" worm which we designed.

In a first scenario we used CMMTool to mimic the behavior the behavior of people inside lunch areas or food courts, creating groups of relatively stationary persons, a small number of which "travel" from lunch area to lunch area. The results are shown in Figure 4. In a first approximation, we did not consider people entering or leaving the shopping mall during our simulation time (on the line marked as "no output"). Then we added a random flow of persons with discoverable devices entering and exiting the mall (on average, one person each 10 seconds, a realistic value from our assessments). In this case, we tested two different conditions: the first is a worm propagating (starting with just one infected device), marked as "no BlueBag" on the figure, and the second is the presence of an attacker with a tool similar to our BlueBag, who is actively disseminating a worm.

As it can be seen, over 30 minutes on average a simple worm would be able to infect any susceptible device in the lunch area, just through propagation: within the time limit of a typical lunch break. An attacker with a device such as the BlueBag would obtain the result even faster.

In a second scenario we considered the behavior of a more mobile crowd of people walking in and out of the shops, looking at the windows. In this case, the results are similar, but they depend heavily on the motion patterns in the

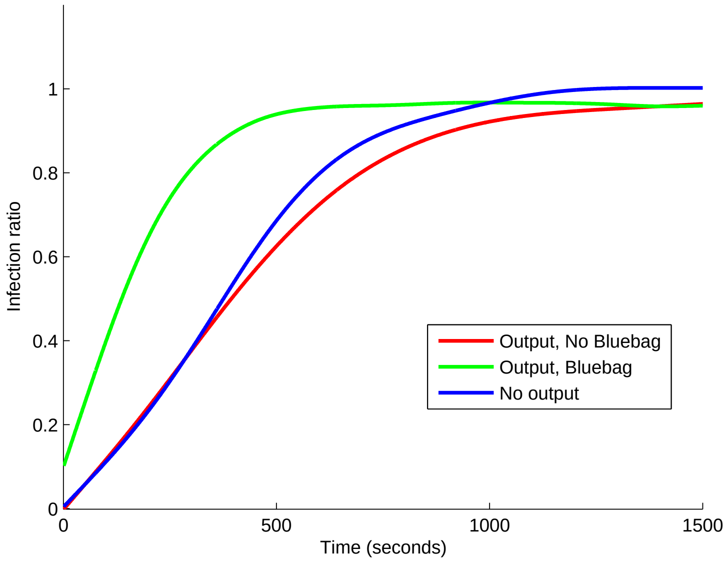


Figure 4: Ratio of infection in our simulation

mall, and are slower than in the food courts scenario (i.e. propagation speed is nearly halved in this case).

7 Conclusions and future works

In this work, we tried to envision possible future scenarios of attack involving targeted malware propagated through Bluetooth enabled devices. During our research, we have created proof of concept codes and devices (such as the BlueBag, eponymous of the research project) demonstrating the feasibility of such attacks. We demonstrated that it is possible to build a distributed network of agents spreading via Bluetooth, which can seek given targets, attack them, and then exploit the devices capabilities to log keystrokes, steal data, record audio data, take pictures, and ultimately send the collected data back to the attacker, either through the agents network or directly.

In other words, we demonstrated how the combination of wide availability of Bluetooth communication and the increasing usage of mobile computing devices creates a new potential risk. The combination of known and unknown implementation weaknesses, which are more difficult to patch than in a traditional computing environment, and of user misbehavior caused by lack of security awareness creates dangerous paths of attacks against data stored in mobile devices.

We also roughly analyzed the effectiveness of such an attack, through real data collected during surveys, conjugated with simulations. We demonstrated that a dense crowd of discoverable, Bluetooth enabled devices exist, and can be easily reached by a worm-spreading attack device. This is due to the joint effect of missing user education and awareness, and poor default configuration choices on the vendors' side.

Possible future extensions of this work include a better planning of the "phone home" payload of the malware, to understand how likely it is for the collected data to reach the attacker under various scenarios and how to improve worm auto execution and process hiding. Also, the creation of a Bluetooth-only command and control infrastructure would be a challenging evolution, since it would integrate Ad-hoc networking issues in our work.

Like common worms, our malware doesn't currently use Bluetooth attacks to spread itself around: in future we want to investigate the possibility to use a sort of attacks library, combining social engineering attacks and Bluetooth technology attacks.

Another possible extension would be the use of the BlueBag as an honeypot, to "capture" Bluetooth worms in the wild and measure their real prevalence. We briefly engaged in this activity, but a more extensive testing is needed to give reasonable statistical results.

Acknowledgments

This work was made possible by a number of key supporters. The early stages of the BlueBag project were jointly funded by F-Secure, a renown antivirus vendor based in Helsinki, Finland; and Secure Network, a security consulting and training firm based in Milano, Italy. One of the authors has been partially supported by the Italian Ministry of University and Research under the FIRB-PERF Project, in the research unit led by prof. Giuseppe Serazzi, whose support we gratefully acknowledge. We thank warmly Martin Herfurt, Marcel Holtman and Adam Laurie, authors of the earliest works on Bluetooth security issues, for their comments on this work. We also wish to acknowledge the help received by Dr. Mirco Musolesi (UCL) and Dr. Paolo Costa (DEI - Politecnico di Milano) on the issue of modeling people movement. A number of persons helped with various stages of this project, including Alvise Biffi, Laura Mantovani, Miska Reppo, Mara Rottigni. We thank all of them warmly. Finally, we wish to thank the anonymous reviewers for their really helpful and extensive review of the first draft of this paper.

References

[1] Trifinite.org website. `http://www.trifinite.org`.

[2] Design and implementation of an enhanced security layer for bluetooth. In *Proceedings of the 8th International Conference on Telecommunications, ConTEL 2005*, volume 2, pages 575–582, June 2005.

[3] Abha Ahuja Craig Labovitz and Michael Bailey. Shining light on dark address space. Technical report, Arbor networks, Nov 2001.

[4] Eric Filiol, Marko Helenius, and Stefano Zanero. Open problems in computer virology. *Journal in Computer Virology*, 1(3-4):55–66, 2006.

[5] Christian Gehrmann, Joakim Persson, and Ben Smeets. *Bluetooth Security.* Artech House, Inc., Norwood, MA, USA, 2004.

[6] S.F. Hager, C.T.; Midkiff. Demonstrating vulnerabilities in bluetooth security. In *Global Telecommunications Conference GLOBECOM '03*, volume 3, pages 1420 – 1424, December 2003.

[7] Herbert W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42(4):599–653, 2000.

[8] Chia-Jui Hsu and Yuh-Jzer Joung. An ns-based bluetooth topology construction simulation environment. In *ANSS '03: Proceedings of the 36th annual symposium on Simulation*, page 145, Washington, DC, USA, 2003. IEEE Computer Society.

[9] Markus Jakobsson and Susanne Wetzel. Security weaknesses in bluetooth. In *CT-RSA 2001: Proceedings of the 2001 Conference on Topics in Cryptology*, pages 176–191, London, UK, 2001. Springer-Verlag.

[10] Jeff O. Kephart and Steve R. White. Directed-graph epidemiological models of computer viruses. In *IEEE Symposium on Security and Privacy*, pages 343–361, 1991.

[11] F-Secure Labs. Wireless threats list. available online at `http://www.f-secure.com/wireless/threats/`.

[12] Robert Morrow. *Bluetooth Implementation and Use.* McGraw-Hill Professional, 2002.

[13] Mirco Musolesi and Cecilia Mascolo. A Community based Mobility Model for Ad Hoc Network Research. In *Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN'06).* ACM Press, May 2006.

[14] K.K. Mookhey Pukhraj Singh. Metasploit framework. `http://www.securityfocus.com/infocus/1789`, July 2004.

[15] Giuseppe Serazzi and Stefano Zanero. Computer virus propagation models. In Maria Carla Calzarossa and Erol Gelenbe, editors, *Tutorials of the 11th IEEE/ACM Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecom. Systems - MASCOTS 2003.* Springer-Verlag, 2003.

[16] Yaniv Shaked and Avishai Wool. Cracking the bluetooth pin. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 39–50, New York, NY, USA, 2005. ACM Press.

[17] Eugene H. Spafford. Crisis and aftermath. *Communications of the ACM*, 32(6):678–687, 1989.

[18] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02)*, 2002.

[19] Steve R. White. Open problems in computer virus research. In *Proceedings of the Virus Bulletin Conference*, Oct 1998.

[20] Stefano Zanero. Issues in modeling user behavior in computer virus propagation. In *Proc. of the 1st International Workshop on the Theory of Computer Virus*, LORIA, Nancy, France, May 2006.

[21] Cliff Changchun Zou, Don Towsley, and Weibo Gong. Email virus propagation modeling and analysis. Technical Report TR-CSE-03-04, University of Massachussets, Amherst.