Sensing-Enabled Channels for Hard-to-Detect Command and Control of Mobile Devices

Ragib Hasan
University of Alabama at Birmingham
Birmingham, Alabama 35294-1170
ragib@cis.uab.edu

Nitesh Saxena
University of Alabama at Birmingham
Birmingham, Alabama 35294-1170
saxena@cis.uab.edu

Tzipora Halevi
Polytechnic Institute of NYU
Brooklyn, NY 11201
tzipihalevi@yahoo.com

Shams Zawoad
University of Alabama at Birmingham
Birmingham, Alabama 5294-1170
zawoad@cis.uab.edu

Dustin Rinehart
University of Alabama at Birmingham
Birmingham, Alabama 35294-1170
drnehart@cis.uab.edu

ABSTRACT

The proliferation of mobile computing devices has enabled immense opportunities for everyday users. At the same time, however, this has opened up new, and perhaps more severe, possibilities for attacks. In this paper, we explore a novel generation of mobile malware that exploits the rich variety of sensors available on current mobile devices.

Two properties distinguish the proposed malware from the existing state-of-the-art. First, in addition to the misuse of the various traditional services available on modern mobile devices, this malware can be used for the purpose of targeted context-aware attacks. Second, this malware can be commanded and controlled over context-aware, out-of-band channels as opposed to a centralized infrastructure. These communication channels can be used to *quickly reach out to a large number of infected devices*, while offering a high degree of *undetectability*. In particular, unlike traditional network-based communication, the proposed sensing-enabled channels cannot be detected by monitoring the cellular or wireless communication networks. To demonstrate the feasibility of our proposed attack, we present different flavors of command and control channels based on acoustic, visual, magnetic and vibrational signaling. We further build and test a proof-of-concept Android application implementing many such channels.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Unauthorized access—*Malware* ; C.5.3 [**Microcomputers**]: Portable devices

Keywords

Mobile security, Mobile Malware, Command & Control, Mobile device sensors, Covert channel

ASIA CCS'13, May 8–10, 2013, Hangzhou, China.

1. INTRODUCTION

The mobile devices have become both "smart" as well as ubiquitous in the recent years. Today's mobile devices, such as smart phones or tablets, are equipped with a multitude of sensors, enabling them to detect their location, and learn the characteristics of their users and the surrounding environment. These rich capabilities have enabled many interesting applications and immense possibilities for ordinary users. However, at the same time, they have opened up the door towards new generation of mobile malware that can exploit the on-board sensors for malicious purposes. Ranging from eavesdropping over the phone call or user input [10, 40, 44], learning user's location [16] to snooping on the user's activities [34], mobile malware can gather sensitive information previously not available to traditional malware.

In this paper, we argue that the sensors present in mobile devices can also be used for out-of-band communication among malware infected devices as well as for targeted command and control. For example, the audio sensors present in a mobile phone can be used to trigger malware located in a specific physical region. Malware can also be triggered or commanded via audio/visual signaling transmitted through television or radio broadcasts. While triggering is currently possible via the cellular or wireless network channels, such messages can easily be detected by monitoring these communication channels either at the mobile phone or at the network gateways. Unlike the traditional command and control communication over a centralized infrastructure (such as a cellular network), out-of-band communication is *very hard to detect* and even harder to prevent. However, it can be still be used to *reach out to a large population* of infected mobile bots.

In addition to the misuse of the various traditional services available on modern mobile devices (such as phone calls or SMS/MMS), we posit that this malware can be used for the purpose of targeted context-aware attacks. For example, a malware that gets triggered in a movie theatre, via say a hidden audio signal embedded in a commercial, can be used for causing annoyance or even chaos; imagine, for instance, the infected devices in the theatre all playing a loud song or a siren suddenly.

Researchers have previously explored various forms of mobile malware and different means of malware communication channels including their detection and blocking. In particular, most of the work has focused on wireless or wired network based command

and control channels over the Internet, wireless, or cellular phone networks. The threat posed by the situational and environmental awareness of mobile malware and the use of the sensors to perform *hard-to-detect out-of-band communication* has not been studied.

1.1 Our Contributions

The contribution of this paper is threefold. *First*, we provide the first detailed study of environmental sensor-based covert channels in mobile malware. In particular, we present different flavors of out-of-band command and control channels based on acoustic, light, magnetic and vibrational signaling. Many of the proposed channels provide a means of *undetectable communication with a large number of malware infected devices* such as through the use of broadcast video or audio signals.

Second, to demonstrate the feasibility of our attacks, we build a proof-of-concept malware application using an off-the-shelf mobile phone on the Android platform implementing many of the proposed channels. We conduct several experiments to validate the effectiveness of these channels for command and control. Some of our experiments are conducted in real-life setting and further confirm the threat posed by the presented mobile malware.

Third, we sketch the possibility of building geographically localized attacks which can leverage upon the aforementioned out-of-band channels.

1.2 Scope and Ethical Aspects

The goal of this paper is to demonstrate the feasibility of out-of-band channels for command and control of the malware. As such, the scope of the paper is limited to the exploration, design and analysis of such channels. In particular, the emphasis is on sending undetectable triggers to infected devices over such channels. Developing models, either theoretical or experimental in nature, to estimate the latency and coverage of such channels (e.g., to determine how many infected devices can get triggered in a given time span) is beyond the scope of this work. However, we emphasize that many of our channels naturally provide rapid message delivery guarantees to a huge number of devices.

Although we are presenting essentially a new generation of attack against mobile devices, the purpose of this work is ethically sound and constructive. By pre-empting the design of this attack and possibly "staying ahead in the game" against the real attackers, our vision is to eventually come up with an effective defense against the envisioned attack. Due to the high level of incentive available to the attacker (high degree of undetectability), these attacks might very well be launched in the wild in the near future. By means of a publication on this topic, we hope to raise awareness about new threats, and motivate fellow researchers, device manufacturers and OS designers to build and deploy defenses before these attacks are launched in the wild. In fact, we also discuss potential approaches to defend against such malware, and aim to further extend them in the near future. Many recently published papers on precisely the same broad topic [8, 13, 19, 26, 37, 42, 46, 49, 50] (reviewed in Section 7) further support and justify this line of security research. Additionally, we are positive that the underlying communication channels designed as part of our research will come handy in other security applications in the future.

2. BACKGROUND AND THREAT MODEL

In this section, we discuss the motivation for out-of-band command and control, and the underlying threat model, and sketch the possibilities for localized context-aware attacks enabled by out-of-band channels.

2.1 Motivation: Why Use Sensors?

Many of today's mobile devices such as smart phones are equipped with optical, audio, vibration, and magnetic field sensors. Some of these sensors, such as accelerometers, are so sensitive that they have been repurposed for distributed sensing applications such as earthquake detection [11]. The optical sensor and the camera present in mobile phones are also becoming more sophisticated day by day. For example, the Apple iPhone 4GS contains an 8 mega pixel digital camera. Similarly, the microphones present in mobile phones are sensitive enough to pick up very subtle sounds from the surrounding environment. For instance, the iPhone 3g's built-in microphone can detect sound as low as 5 Hz to as high as 20 KHz [2]. Many devices besides smart phones, such as tablets and laptops, also come equipped with many of these sensors.

The sensors can serve as an appealing platform for out-of-band communication among malware infected devices as well as between the botmaster owner and infected devices. Unlike the traditional centralized means of communication for malicious purposes, such an out-of-band communication can remain *very hard to detect*, especially if covert and steganographic communication channels are used. However, it can still be used to communicate with a large number of devices, which can even span international borders. For example, messages sent embedded within the audio of a popular TV program can be delivered to a huge number of infected devices, whose users would be watching such a program.

Easy Detectability of Network-based Channels: For the botnet communication/triggering described above, a traditional network-based channel, e.g., a TCP/IP based channel, is easily detectable. TCP/IP-based triggers can be detected and/or blocked by a firewall or anti-malware software which monitors network packets. Applications constantly polling incoming packets, registering for push notifications, or accessing unknown web services can raise suspicion. Even triggers steganographically hidden in benign communication can still cause suspicion: why would an app run even benign protocols with random servers (other than known backends), assuming the botnet controller has not compromised legitimate servers?

Unique Advantages of Sensor-based Channels: The detection of out-of-band signals is complicated by the fact that the out-of-band trigger signal format can be free form. Traditional botnet command and control messages, on the other hand, travel over centralized networks obeying established protocols such as UDP or TCP/IP. However, the out-of-band covert channels can use arbitrary protocols to send the control and command messages. This makes the detection of such communication quite difficult in practice. Anti-malware software also does not know which medium is used by the botnet for covert communications: it must therefore monitor all sensors constantly. Also, accessing a sensor is not necessarily the "signature" of a malware: many legitimate applications need access to sensors for benign purposes.

Sensor-based trigger channels have other useful properties. For localized attacks, sensor channels are more effective than TCP/IP. In an area, not all phones may use the same network or have network connectivity, but most phones there share the same medium. Also, without knowing infected devices' IP addresses, the botnet controller has no way to trigger them (other than broadcasting/flooding entire network, which is infeasible), whereas in sensor-based channels, the controller does not have to know any addresses of infected devices.

2.2 System Model and Assumptions

Our system model is *no different* from the model employed in traditional command and control of malware. Namely, we assume that many mobile devices have already been corrupted with mal-

ware. Such corruption could take place, for example, when the user downloads an untrusted application – embedding the malware – from the application store of the service being used. However, to remain surreptitious, the malware on these devices will not activate or get triggered until indicated by the botmaster. These triggers will later be sent by the botmaster to (all or a subset of) infected devices over out-of-band channels. Such channels can also be used for the purpose of sending commands to the bots. However, in this paper, our primary emphasis is on triggers.

We assume that the (malware) application is allowed to run in the background and can access the device's on-board sensors without restriction. Indeed, as our prototype implementation shows, the Android platform supports such apps when using a microphone, a light sensor, and a magnetometer, as well as an accelerometer. Other operating systems, such as iOS, may not conform to this policy. However, it is also possible for the malware to attach itself to a benign app that needs constant access to sensors as a background process (e.g., a web search application that uses voice input).

Additionally, we assume that, with a very high probability, the mobile devices are switched on and are in close physical proximity of their users, either carried by the users in their pockets, purses or backpacks, or lying close to them. For example, while watching TV, a user's phone is placed next to him/her. Given that modern users heavily rely upon their mobile devices (especially phones), this is a valid assumption to make [17, 18, 24, 25, 29, 31, 33, 41, 45, 47, 48]. Under a rare circumstance that a phone is powered off or not close to the user, the messages can not delivered to that particular phone at that particular time. This would only degrade the overall reachability of the messages slightly.

Once activated by a trigger, the infected devices will carry out the attacks they are programmed for. Traditionally, these attacks could be used for the purpose of spam campaign, making illicit phone calls or sending SMS/MMS messages. In addition to these attacks, we envision another genre of attacks, which we discuss in the following subsection, that can be used for targeted and localized threats.

2.3 Localized Targeted Attacks

A localized mobile botnet consists of malware infected mobile devices physically present in a specific location. An attacker can launch localized attacks on or through these devices. For example, during a sports event, an attacker can trigger the malware-infected mobile devices in the sports arena. To do so, the attacker issues command and control messages broadcast via out-of-bound channels.

We outline following types of active context-aware attacks. Validating the feasibility of some of these attacks is beyond the scope of this paper, however.

Distributed Denial-of-Service Attacks: The infected mobile devices could be used to launch localized denial-of-service attacks on a certain network. For example, the infected devices present in an airport may be "commanded" to collectively bring down the airport's WiFi network.

Annoyance Attacks: The malware on the devices could be used to cause annoyance or even chaos in a public place. E.g., in a movie theater, an advertisement may be used to send covert triggers to all infected phones present in the theater; and then the malware on one phone may interact with other phones (say make phone calls to one another). The malware may also collectively play some loud music or sirens. Given that many people do not "silence" their phones in a theatre, such an attack can certainly cause a lot of annoyance.

Embarrassment Attacks: Selectively triggered malware could be used to cause embarrassment to the device's user or those present nearby. For instance, a person may be using her (infected) phone to project a presentation in a conference. As the person starts to speak, another infected phone in the room can trigger the phone malware (via some out-of-band channel), which would then project an embarrassing video onto the screen.

Safety Hazards: Public safety hazards are also within the purview of context-aware malware. As an example, the malware on the phone can be triggered when the infected phone is inside a driving car; the malware may then interact with the car's internal network and cause some serious problems. Similarly, a malware may get triggered inside a home/company and may then interfere with the home's wireless security system – perhaps dismantle it. This will clearly prompt the possibility of theft or burglary and may endanger the lives of the inhabitants.

Interference Attacks: Context-aware malware may be used to cause interference with the surrounding environment. For example, the infected mobile devices can selectively interfere with an aircraft radio system at the time of take-off or landing, or with the medical devices in a hospital [9, 30].

Distraction Attacks: Another viable and perhaps very interesting attack would be a "distraction" attack. Here, the malware aboard a user's mobile phone tries to distract the user while she is performing a security task (e.g., reading a security warning; pairing her devices, etc.). For example, the phone may play a ring tone or vibrate to distract the user. It would be interesting to see how users fair under such a distractive attack.

3. OUT-OF-BAND COMMAND AND CONTROL OVERVIEW

In this section, we provide an overview of different out-of-band channels using which a botmaster can trigger the infected devices. A detailed description of these channels and their various characteristics appears in the following section.

We divide these channels into two categories, one of which explicitly uses steganographic techniques to make the task of detection extremely difficult. The receiving side of these channels can be built using the following sensors on mobile devices:

- Audio sensor / microphone
- Camera / light sensor
- Magnetometer
- Accelerometer / vibration sensor

3.1 Steganographic Channels

A steganographic channel is one where the trigger signal is hidden inside another signal. This channel would usually involve the trigger being embedded inside a song or other audio. Trigger signal broadcasts using such a channel can be achieved either via audio signal embedding or live stream embedding, as discussed below.

Audio Signal Embedding: In this variant, the audio signal is embedded inside a recorded carrier audio which is later broadcast using TV or radio. Here, the attacker does not need to be physically present near the target. The trigger is embedded in an innocuous video or audio (e.g., a song or a TV program). When the program is broadcast through television or radio, any malware infected device in the physical proximity of a running television or radio will receive the signal. Other mechanisms, such as embedding the trigger signal inside music played by musical greeting cards, can also be used to spread the trigger.

Live Stream Embedding: In this variant, the carrier signal or audio, containing the trigger signal, is broadcast or played directly near the target mobile devices. The trigger signal can also be embedded in broadcast video or audio signals in a different manner. During live telecast of an event (such as a game or a speech), the attacker can

play the trigger-embedded audio near the microphone used for the broadcast. The microphone can pick up the infected audio stream and thereby broadcast it to a large audience.

The attacker can also tap into the broadcast workplace music delivery networks and embed the trigger into the music. For example, background music services such as Muzak [38] deliver music to workplaces, hospitals and elevators, which may be leveraged for this purpose.

3.2 Non-Steganographic Channels

A non-steganographic triggering channel is one where the trigger signal is not hidden per se, but rather it is delivered directly.

Audio Patterns: The malware can be programmed to be triggered by a specific audio pattern (e.g., a song).

Ambient Light: The attacker can tap into the power supply of a building and cause rapid fluctuations in voltage, resulting in rapid but imperceptible flickers in the lights all across the building. A trigger message embedded via such flickers can be read by any infected phone in the building.

Magnetic Signalling: The attacker can induce a strong enough magnetic field and send the trigger by changing the strength of the field. Using this scheme, the attacker can hide a magnetic field inducing device in a crowded area, as an example, and trigger the devices whose users pass by the magnet.

Vibrational Signalling: Attackers can also use vibration channels. For example, messages can be encoded into vibrations produced by a subwoofer which can be read by nearby phone accelerometers.

4. CHANNEL CHARACTERISTICS

For transmitting command and control messages to mobile devices, we examined the properties of audio, light, magnetic field and vibrational channels overviewed in the previous section. In this section, we discuss the various properties of each of these channels including: *range*, *noise characteristics*, *adversarial control*, *coverage and reachability*, *latency*, *bandwidth*, and *steganographic capabilities*. A summary of this analysis is depicted in Table 1.

4.1 Audio Channel

An audio channel is composed of an audio signal, which is used to encode the trigger message. An audio channel can be steganographic or non-steganographic, as mentioned before. In the steganographic mode, the trigger signal is embedded in a song or other music/audio signal. In the non-steganographic mode, the audio signal is used directly.

The attacker can use a TV or a radio program, background music services, an internet TV program and musical greeting cards to transport its commands over the audio channels. When using a TV or radio program, the attacker would either need to manipulate an existing program (such as a popular show or a commercial) or insert a live audio stream at the recording of a live event (such as a sporting event). While the former capability may require the attacker to collude with an insider at the service itself, the latter capability only requires the attacker to hide a audio transmitter near the recording station. The attacker may also register its own commercial with the service within which it can insert the audio commands. Although this incurs a cost to the attacker, it offers the advantage of covert communication that can be broadcast to a large audience, possibly multiple times, in a day. Similarly, when using a musical service, the attacker would need to tamper with an ongoing music broadcast. This may again require collusion with an insider at the music service or hack into the inside network of the service.

In contrast to the traditional TV, radio or musical services, internet TV – especially P2P-TV [4, 5] – appears to be more lucrative for the attacker. Using such a channel, the attacker can simply send out its commands embedded within its *own program* (i.e., for which the attacker itself is the source of the streaming data). Since P2P-TV is completely decentralized and any user can become the source/receiver of a program, the attacker can very easily communicate with the infected devices through this mechanism. It is also possible for the attacker to insert its commands within an ongoing (popular) program as demonstrated in an attack [12] on a popular Chinese live streaming program called PPLive [4].

All of the above approaches provide broad coverage and reachability. The TV or radio programs can be used to communicate with all infected devices whose users are tuned to these programs (assuming that the devices are close to their users). The same applies to the Internet TV programs; here the users would be watching the programs on their computers. When using popular programs or live events in each of these settings, the attacker can ensure that its commands can be delivered to an extremely large number of infected devices across the globe. In the case of the workplace music setting, the commands can be received by the infected devices of all users present in the given building (including visitors). This may cover an enormous number of devices especially when a big hotel, library, or an airport, or perhaps many of them, is/are used as the attack target. With musical greeting cards, the commands can be delivered to any infected device whose user opens up or plays the cards. The reachability of this attack clearly depends upon the number of cards dispatched by the attacker. However, one can imagine a broad coverage especially during peak holiday seasons when users actively select/test greeting cards (and play them) at local stores. Those receiving these cards as gifts from others are also reachable because they will also be playing the music.

Another important measure of the effectiveness of command and control channels is their latency, i.e., the delay incurred in delivering the messages to infected devices. All of our settings introduce little latency and the delivery of messages can take place as soon as the users tune to the given program, or are present in a given building. Thus, the delay is only up to the broadcast of the program. With greeting cards, the delay is up to the point users open up and play the cards.

When looking at the audio channel properties themselves, the range and distortion might be two limiting factors. An audio channel must use audio signals that are strong enough to traverse the distance between the trigger transmitter and the mobile device. In addition, background noise often tends to drown the source audio signal. Luckily, in our attack setups, the desired communication range is short, less than a few feet at most. The Internet TV set-up is again the most promising in terms of the range since users' devices are expected to be only a few inches, or about a feet, away from their computers playing the program. Greeting cards also require similar ranges. TV programs, on the other hand, will usually require a communication range of several feet, given that the ideal TV viewing distance is several feet (depending upon the size of the TV screen) [6]. This, however, will be compensated by the powerful speakers of modern TV sets or their associated home theatre systems.

The sensitivity of mobile device microphones can also impact sound reception. Mobile phone microphone are reasonably sensitive (as discussed in Section 2.1), but most mobile operating systems perform noise cancellation of some form at the driver level. Hence, the audio received by a mobile phone may sometimes be not exactly the same frequency as the original sound. The audio channel can also be distorted if the target mobile device is inside pockets or purses.

Mode of Transport	Audio				Light	Magnet	Vibration
	TV / Radio Program	Music Services	P2P-TV Program	Musical Greeting Cards			
Targeted Devices	Any mobile device with a microphone (all phones, many laptops, tablets)			Any mobile device with a microphone and carried by the users (all phones)	Any device with a light sensor or camera (many phones, laptops or tablets)	Any device with a magnetometer (most smartphones)	Any device with a vibration sensor or an accelerometer (most smartphones, many laptops / tablets)
Attacker Control	•Insert audio at the recording of a live event •Manipulate an existing program or commercial •Register its own commercial	Manipulate an ongoing broadcast	• Stream its own program • Manipulate an ongoing program	Dispatch its own greeting cards	• Insert visual fluctuations in an existing TV program or commercial • Registers its own commercial • Tap into the power line	• Hide a magnet in a crowded area • Carry a magnet in a pocket or backpack; walk beside users in a crowded area	• Bring vibrating device close to the users • Induce a high bass sound in a TV or radio program
Coverage and Reachability	All users tuned to the program	All users in the building where the music is played	All users tuned to the program	All users who play /open the cards	• All users tuned to the TV broadcast • All users present in a given building	All users who pass by the magnet	All users who are in physical proximity of the vibration
Latency	Up to the telecast of the program	Up to the broadcast of the music	Up to the telecast of the program	Up to the propagation of the cards	Up to the TV telecast, or flickering of the lights	Up to the users' passing by the magnet	Up to the users' sensing vibrations
Steganographic?	Can be				Can be in the case of a TV broadcast	No, but imperceptible	Can be (imperceptible)
Desired Range	Several feet	Several feet	A feet or so	A feet or so	Several feet	Few centimeters	Several feet with strong subwoofers; else few centimeters
Presence of Noise	Noise is common				Noise not common	Little noise	Noise is common
Works when device is stowed?	Not always				No	Yes	Yes
Expected Bandwidth	Few bits per second						
Peer-based propagation?	No				No	No	Yes

Table 1: Comparison of different out-of-band channels for command and control

Furthermore, audio noise is fairly common and can complicate the task of audio decoding. Noise is quite common in public places.

The bandwidth of the command and control channel may also be important in some scenarios, especially if malware payloads or new malware programs are to be disseminated to the infected devices. The use of out-of-band channels in general, however, exhibits low bandwidths due their fundamental physical characteristics. The audio channel bandwidths are expected to be limited to only a few bits per second [32]. This is an obvious trade-off in using out-of-band command and control in contrast to the traditional channels – the former provide better undetectability while the latter possess better bandwidths. However, even with low bandwidth, the envisioned audio channels can definitely be used to rapidly send triggers to a large number of devices. In addition, to transmit larger messages or other data, the audio signalling could be spanned over the entire program (such as a game or a movie), or multiple programs, or multiple occurrences of the same program.

4.2 Light Channel

The light channel uses the ambient light sensors present in most smart phones and many laptops/tablets. These sensors are very sensitive to the ambient light. The intensity of most light sources depend on the voltage of the power supply. Thus, by changing the voltage, it is possible to modify the ambient light in an indoor location. An attacker using a light channel to broadcast the messages can tap into the power supply of the building and introduce fluctuations, by modifying the voltage or by introducing very short flickers. All rooms in the building will be affected under such an attack. Getting access to the power supply of a building can be a formidable obstacle for the adversary, although such attacks have previously been reported in the wild [36]. However, the payoff in this case is higher: by tapping into a single place, the adversary can cover all areas of a building. Another possibility is to embed the messages within the light variations into an existing or attacker chosen TV program or commercial, similar to some of the audio-based approaches.

Similar to audio channels, light channels also open up the possibility of a broad coverage. When using a TV broadcast, messages can

reach all infected devices of users watching the broadcast (assuming the phones are near their users and not stowed inside pockets/purses). This is especially true when large screen TV sets are used such as in a pub or night club.

These channels also incur minor latencies. These are limited to the point the TV broadcast is screened or the lights are flickered in the building. In terms of undetectability, the flickering of the lights can be so quick that they are not perceptible by the humans present in the surroundings. The TV broadcast can be explicitly made steganographic similar in spirit to the audio based stego broadcast.

As far as the range of the light channel is concerned, several feet is easily workable. This clearly covers the indoor environment such as the scenario of watching a TV screen and fluctuating lights mounted on low ceilings.

The light channel, however, exhibits some limitations. First, to be able to receive a message via the light channel, the mobile device's light sensor must be unobstructed. If the mobile phone is kept inside a pocket or purse, the light channel can not be used. However, many computing devices other than mobile phones – such as tablets or laptops – also contain cameras or light sensors that can be used to measure the intensity of ambient light. The light channel can be effectively used on such devices as they are not usually stowed when powered on.

Another challenge with the light channel is the presence of noise. Ambient light sensors usually do not distinguish between light from different sources as long as it falls inside the visible light spectrum. Therefore, the noise in a light channel can be quite high if there are other frequently changing light sources (e.g., multiple TV sets displaying different programs in a pub) in the same indoor space. In contrast to the audio channel, however, the likelihood of the presence of visual background noise is much less, and thus the light channel is expected to be more robust in practice.

The light channel is expected to provide similar levels of bit rates to that of the audio channel for data transmission.

4.3 Magnetic Channel

The magnetic channel uses the magnetic field sensors (magnetometers or compass) present in most smart phones. Here, the attacker can place small electromagnets in a crowded area. The electromagnets can be controlled by a self-contained device, or via commands transmitted to it from the botmaster over a wireless or cellular network or text message [46]. The electromagnets are used to create a variable magnetic field. The trigger message can be encoded within the presence or absence, or a range of threshold values, of the magnetic field strength.

Magnetic fields dissipate quickly and are inversely proportional to the cube of the distance between the magnet and the device [15]. Thus, a challenge is to create a magnetic field strong enough to be detected above earth's natural magnetic field (the background field). Generation of such a magnetic field is significantly difficult and requires very high electric currents.[1]

However, the distance issue can be resolved by clever placement of the magnetic transmitter. For example, the transmitter can be placed in an elevator of a building or an entrance door frame of a popular building (a hotspot). In such a scenario, the mobile devices carried by people who walk past such transmitters can be triggered. It is also possible for the attacker to physically carry this magnetic device in a pocket or backpack and deliberately walk past users in a crowded area (such as a subway, mall or a sporting event).

Another scenario where the magnetic channel will prove beneficial to the adversary is the usage of Near Field Communication (NFC) phones. Here, the adversary can simply hide a magnet on the NFC readers used for payment transactions. As a user brings his/her phone close to the readers for making the payment, the phone will receive the message.

Compared to the audio and light channel, the reachability of the magnetic channel is a bit limited, and is restricted to triggering devices in the close vicinity of the magnetic source. However, it could still be leveraged to deliver messages to a significant number of devices especially in a crowded region or during rush hour. Its latency is up to the point users come in close physical proximity of the magnetic source.

A distinctive advantage of the magnetic channel over its counterparts is that it requires no line of sight and can work even when the devices are stowed inside pockets or purses. In fact, except of iron and steel, other materials have almost no effect on shielding magnetic fields. Another advantage of this channel is that it is least affected by the background noise (such as that introduced by other magnetic devices or the Earth's field) as long as a high enough detection threshold is used.

4.4 Vibrational Channel

In the vibration channel, the transmission is achieved by inducing vibrations which are then read by a vibration sensor or an accelerometer present on nearby devices. At first, it may seem difficult to induce vibrations to form such channels, but there are (at least) two possibilities for the attacker. One is to embed the messages as high bass sounds into a TV or radio program. Assuming that a subwoofer system is used as the audio transmitter connected to the TV or radio receiver (as in a home theatre or movie theatre system, for example), the associated vibrations can travel some distance and reach the nearby devices. The accelerometer on-board such devices can then decode the vibration patterns. This is especially true when using strong sub-woofer systems as well as buttkickers and vibrating seats already present in many home and movie theatre systems [1].

Another possibility is a peer-to-peer based gossiping channel whereby a vibrating mobile device transmits messages to a nearby device, which then vibrates in turn and transmits to its neighbors, and so on. Such a channel can be formed, for example, in a conference room where many people share a common surface (table) through which the vibrations can travel from one device to the other. This channel is quite feasible and inline with a recent work [34] which demonstrates that keystrokes on a laptop can be learned via an accelerometer on a nearby phone.

The subwoofer based TV broadcast channel provides the same level of coverage and latency as that of the TV based audio channels. The gossiping based channel, on the other hand, has limited reachability and relatively high latency due to its proximity requirements. However, it is highly undetectable due to its decentralized propagation. Both of the channels are also naturally imperceptible to the users. The TV based channels can be explicitly designed to be steganographic just like the audio.

The range of the TV based channel can be up to several feet, especially while using vibrating seats. The gossiping based channel is clearly restricted to a few centimeters. Both of these channels are affected by the background vibration noise, such as that caused by a subtle movement of the users themselves. They are also expected to provide low bandwidth, perhaps less than what is provided by the audio channels.

[1] According to the Biot Savart's Law [15], it will take a wire carrying a large amount of current (500 amperes) to generate a magnetic field strength of just 100 microtesla even from a distance of 1 m. A 500 ampere current will be impossible to induce; as a reference, a current of about 1 ampere can cause electrocution.

5. DESIGN, IMPLEMENTATION AND EXPERIMENTS

In the previous section, we discussed a wide variety of out-of-band command and control mechanisms. To demonstrate the feasibility of mobile malware triggered by covert channels, we have designed and implemented a selected set of these channels. These are listed below:

- Steganographic audio channel
- Direct (non-steganographic) audio channel
- Ambient light channel
- Magnetic-field channel

We have left out the feasibility analysis of vibrational channels from this paper, given that it is expected to exhibit more or less the same characteristics as the audio channel. This is an interesting item for future research, nevertheless.

5.1 Prototype Applications and Test Device

To evaluate the viability of sensor-based covert channels, we developed a set of applications on the Android 2.3.3 (Gingerbread) platform. Android provides support for different types of sensors needed as part of our channels, including the microphone, ambient light sensor and the magnetometer. We ran the prototype applications on an HTC Evo 4g smart phone. These applications ran in the background as Android services. The reported bandwidths in the following experiments are the maximum bandwidths achieved under experimental conditions.

5.2 Audio Channel Design

Direct Channel: When implementing the direct audio channel, we used frequency modulation to encode the messages. The audio signal was created using a 17 KHz carrier signal. The data transmission rate of this channel was 1 bit/second.

Steganographic Channel: Audio Signal Embedding: To develop this channel, we created an embedded signal based on replacing certain audio frequencies, similarly to the idea introduced by Gopalan et al [14]. We utilized two frequencies to encode our data: 1500 or 3000 Hz. If the embedded bit value is 0, then the leading frequency is 1500 Hz, otherwise, it is 3000 Hz. For each bit, the leading frequency power is set to 0.25 of the total frame power. The power of the other frequency is set to 0.001 of the total frame power. We divided the signal into frames of 0.25 sec each. Each frame is utilized to embed one bit. To obtain the frequency spectrum of the frame, we calculate its Fast Frequency Transform (FFT). The frame power is calculated as the summation of the square values of the absolute frequency coefficients.

We initially attempted to utilize only one frequency coefficient to embed each bit. However, this did not produce decodable results. To improve performance, instead of setting just one frequency component (closest to 1500 or 3000 Hz), the 3 frequency components closest to the leading frequency were set to the appropriate power. To detect the beginning of the signal, we added at the beginning of the audio a hail signal. The signal is a perfect sinus at 4500 Hz and lasts for 0.1 sec.

To decode the signal, we first produce a perfect hail signal of 0.1 sec and correlate it to the recorded signal using a 0.1 sec step to advance. We normalize both the hail signal and the inspected signal by setting the signal power of each segment to 1. We then calculate the correlation between the normalized hail signal and the inspected signal. If the correlation is above a certain threshold, we know we reached the region of interest. Once the region of interest has been reached, we further advance to the consecutive segments until the maximum correlation is achieved. At this point, we reduce the step size to 0.01 sec and examine the surrounding 0.1 sec around the found location until the updated maximum correlation is achieved.

At this stage, we decode the rest of the signal. We examine each 0.25 sec stream and calculate the FFT for it. We then calculate the power in the 3 frequencies closest to the frequencies of interest (1.5 KHz and 3 KHz). We divide the power of the first frequency coefficients set by the power of the second frequency coefficients set. If the resulting value is above a certain threshold, the value is marked as 0; otherwise, it is marked as 1.

Steganographic Channel: Live Stream Embedding: In this scenario, we add a signal which embeds a random bitstring to an unknown live stream. We create a high-frequency signal for each "mark" bit (a bit with value 1). We then play this signal together with the streaming signal and decode the result. We create a hail frequency with length of 0.1 sec and frequency of 4500 Hz at the beginning of the signal. For the "mark" bit, we utilize the 20K, 20.1K, 20.2K up to 20.5K frequencies. We create for each of these frequencies a perfect sinus and add all of them to create a combined signal of length 0.1 sec. To prevent "stretching" of the signal we further add a 0.1 sec of break after the mark. For the 0 bit, we leave a 0.2 seconds break. We play the resulting key audio stream together with the original stream. We then record the resulting audio and decode it.

To decode the signal, we first detect the hail frequency (in the same way as in the case of data embedded in a known signal). We then calculate the FFT for each 0.1 signal segment and add the sum of frequencies in the 20 - 20.5 KHz. We use a threshold on the sum to detect the value of each bit.

5.3 Audio Channel Experiments

5.3.1 Direct Channel

We conducted our experiments involving the non-steganographic or direct channel keeping in mind the various attack scenarios involving the audio channel: TV/radio broadcast, background music services, IP-TV program, and musical greeting cards (outlined in Section 4; summarized previously in Table 1). To this end, we decided to use a pair of low-end computer speakers as our audio transmitter. (A picture of our experimental set-up appears in Figure 4, moved to the appendix due to the space restrictions). This choice was made to simulate the transmission of audio from an average laptop (e.g., in the case of an IP-TV program), wall-mounted speakers (in the case of workplace music services), or even a musical greeting card. When sending audio via a TV set or a home theatre, much powerful speakers are used in practice. As our experiments below demonstrate, even with low-grade speakers and low volume levels, we were able to achieve reasonably long communication ranges.

Indoor Tests: We first tested the application inside a room (10 ft by 10 ft), with the phone placed on a desk. In this setting, we found that we can successfully send the trigger message to the phone without any data loss (i.e. 0% bit loss) from one side of the room to the other. This validates that the audio triggers can be effectively transmitted to phones, via the speakers of a desktop or laptop computer, in a small personal office setting.

Next, we tested the application in the hallway of a building. In this setting, there were a number of factors introducing environmental noise, such as people talking, kitchen appliances (hallway was alongside a kitchen), and air conditioners. We placed the speakers at one end of the hallway, and the phone at varying distances from the speakers. Despite the presence of "real" background noise, we were able to send the trigger message up to 55 feet distance (the hallway length) with no transmission errors or bit losses, even when the speaker volume was kept low.

To evaluate the effect of loud background noise, we next embedded the signal in a music video. We found that even this large noise introduced by the music video does not affect the transmission rate nor does it introduce any bit errors, up to the distance we tested (55 feet).

These experiments provide evidence that triggers can be easily dissipated via the speakers of a TV set even with large viewing distances, such as at a home, in a movie theatre or a pub with big TV screens. Given that these speakers under realistic conditions would be much powerful, the triggers can be sent, if needed, over much longer distances than the ones we tested. To conclude, the range of the audio channel can be much higher in practice, contrary to intuition.

Finally, we evaluated the scenario where the phone is kept inside a pocket or a purse. We found that the placement of the phone under two layers of clothing (a T-Shirt wrapped all over the phone, including its microphone) did not affect the reception of the trigger message from different distances. The phone was also able to receive the trigger even inside a purse when tested at a distance of up to 22 feet from the speaker. This suggests that audio triggers can be sent even when the phones are stowed inside users' pockets or purses. This ascertains specifically the possibility of transmitting triggers via musical cards in a store. Although we did not test the effect of keeping the phone inside small leather covers (which some people do), we believe that such materials might be able to completely shield the audio signals. In such cases, the triggers may not be received.

Outdoor Tests: We repeated our experiments outdoors in the presence of large environmental noise. We used the same set of low-end speakers to transmit the signal. Despite the large amount of noise due to passing by vehicles and people, the application was able to receive the trigger signal at a distance of 45 feet, with no bit errors. In case of the signal being embedded in a music video, the transmission range was reduced down to 25 feet.

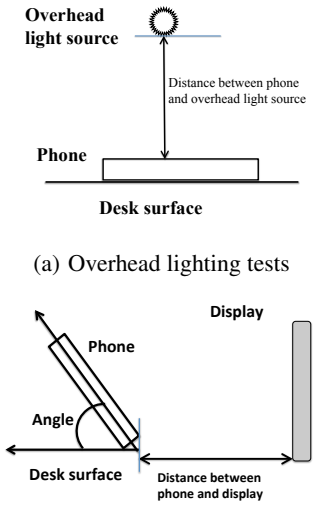
This test indicates that the audio channel can be quite effective even in a realistic outdoor environment. Thus, it is not hard to imagine sending triggers to a large number of phones in a localized area such as a football stadium.

5.3.2 Steganographic Channels

For evaluating our steganographic channels, we further conducted some experiments in a office room setting. To test our algorithm for the embedded audio signal channel, we took a sentence from a wav file (from a James Bond movie), and used the encoding to insert a random 24-bit message in it. The audio was then played and recorded. The signal was decoded successfully and the correct key was detected. We ran the tests from close distance (using a laptop's built-in speaker). We also ran the tests from 3 ft away using a desktop with a speaker built into its monitor. We found the detection was successful from both distances. The sampling rate for the recording was 44.1 kHz. The transmission rate for this channel was 4 bits per second. The detection accuracy was reduced slightly when moving to a farther distance, achieving a 92% success rate at 6 ft (22 bits out of 24 bits successful) and 66% detection rate at 8 ft (16 bits out of 24 bits successful).

In case of the live stream channel, we encoded a random 128 bit long message. We played both the encoded message and a song in the background ("at last" by Etta James). Our tests showed that the entire message was decoded successfully. This channel had a bit rate of 5 bits per second. This was further tried from distances of 3, 6 and 8 ft, and the message was decoded successfully for all these distances.

The above result indicate that even steganographic channels can be effective at least in a non-noisy environment. Further tests are needed to evaluate these channels under larger amount of noise both indoor and outdoors.



(a) Overhead lighting tests

(b) Computer and TV lighting tests

Figure 1: Set-up for the light channel tests

5.4 Light Channel Design and Experiments

To evaluate the ambient light channel, we explored the effect of signals sent through two light sources: (1) overhead lighting, and (2) different types of computer and TV screens located near the phone. This captures the attacks whereby the triggers are sent via power fluctuations, or via TV/IP-TV programs. Figure 5 in the appendix shows a snapshot of our experimental set-up.

For this channel, we assumed that the mobile phone will be placed on top of a table. As mentioned earlier, we designed our malware application as a background service in this case. We note that the typical ambient light inside a room during daytime is between 120 and 240 lux units. At night, with a fluorescent light, the ambient light is about 120 lux units.

Overhead Lighting Tests: When placed on top of a desk, a mobile phone is usually about 6 feet away from the ceiling, which is where the light sources are commonly located. To simulate this scenario for the overhead light channel tests, the phone was placed flat on the desk in a office room, with the display facing up (the ambient light sensors of most smart phones are located at the top of their displays). Figure 1(a) depicts this set-up. We manually introduced small fluctuations in the light intensity of the overhead lights inside the room. Our experiments in this setting showed that by tweaking the light source, we were able to change the light intensity from 120 lux to 160 lux as detected by the phone's ambient light sensor. In this setting, we were able to send the trigger message to the phone successfully with 0% bit errors at a data rate of 0.5 bits per second.

Computer and TV Screen Tests: Next, we examined the feasibility of a signal being sent via a video played on a computer monitor or a TV. Figure 1(b) shows our experimental setup for this set of tests. Here, we varied the tilt angle between the phone and the desk, and measured the maximum range with 0% bit errors. The trigger signal was displayed on the monitor. To encode the messages, we programmatically increased and decreased the brightness of the screen.

The total amount of light emitted by a monitor depends on the size of the monitor. For our tests, we used a 17 inch laptop display, a

Angle (degree)	Bit Rate (bits/second)	Max Distance (inch)
90	1	3
90	0.5	65
60	0.5	16
45	0.75	5
45	0.5	11

Table 2: Results of the Laptop screen experiment

22 inch desktop LCD display, and a 48 inch LCD television. In case of the laptop and the TV, we performed the experiments at night with the overhead lights turned on, providing bright ambient light in the room. In case of the desktop screen, we performed the tests during the day (i.e., in presence of natural light) with a few overhead lights on.

Table 2 depicts the results we obtained when using the laptop screen. The angle between the phone and the surface was varied from 90 degree to 45 degree. We were able to sustain a bit rate of 0.5 bits/second at a distance of 65 inches or 5.5 feet with 0% bit loss when the phone is parallel to the laptop screen. As the tilt angle is lowered w.r.t. the desk surface, the range decreases. In summary, we achieved reasonable range even with a small screen monitor.

Table 3 shows the results of our tests when the desktop monitor was used. The results in this case are similar to the previous test with the laptop monitor.

Angle (degree)	Bit Rate (bits/second)	Max Distance (inch)
90	1	7
90	0.5	24
90	0.33	35
60	0.5	15
30	0.75	6
10	0.75	4

Table 3: Results of the 22 inch LCD desktop experiment

We found the best results (longest range) in the case of the 48 inch LCD TV. Results from our TV experiment are shown in Table 4. Even with the phone placed flat on a desk, we received the trigger message with 0% bit errors at a distance of 13 inches from the TV. We were even able to receive the signal at a distance of 100 inches (8.33 feet) when the phone was parallel to the TV screen. This clearly demonstrates the effectiveness of our light channel for sending triggers via TV broadcasts. Note that the normal TV viewing distances [6] are inline with the range we obtained based on our experiments.

5.5 Magnetic Channel Design and Experiments

For the magnetic-field channel, we built a programmatically controlled electro-magnet to encode the messages. At the receiving end, we retrieved the readings of the magnetic sensor of the phone via an application. The magnetometer built into the phone is a 3-axis vector magnetometer and thus provides the X, Y, Z components of the magnetic field relative to the spatial orientation of the phone, as shown in Figure 2. Choosing only one component of the magnetic field would require the device to be held in a particular orientation. Instead, a scalar value makes use of all three component values and allows the device to be in any orientation. We can simply derive this scalar value by using the formula for vector-to-scalar conversion: $\text{scalar} = \sqrt{m_x^2 + m_y^2 + m_z^2}$, where x, y, z denote the X, Y, Z components of the magnetic field, respectively.

We first calculated the background magnetic field due to the Earth's magnetic core. We turned on the magnetic sensor on the phone and recorded the scalar values to measure the ambient magnetic field strength at various locations in the absence of a strong magnetic source in order to determine this baseline. The values we obtained fall within the range of 30 to 50 microtesla, which

Angle (degree)	Bit Rate (bits/second)	Max Distance (inch)
90	1	18
90	0.5	100
30	0.75	46
15	0.75	28
0	0.5	13

Table 4: Results of the 48 inch LCD TV experiment

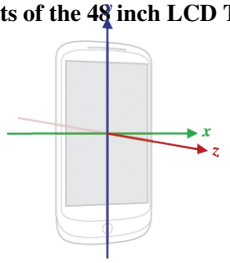


Figure 2: The reference axes across the phone

conforms to the fact that the Earth's magnetic field at the equator is 31 microtesla.

Next, we set out to test two different scenarios. First, when the (electro-)magnet is not covered, and second, when the magnet is covered by clothing, plastic and other material. This was done to study any potential effect on the magnetic field strength due to hiding the magnet on a door frame. For each scenario, we varied the orientation of the magnet with respect to the phone. A snapshot from our experiments is shown in Figure 5(a) in the appendix.

Figure 3(a) shows the various distances, between the phone and the magnet, at which the presence of the magnetic field can be *detected*. Notice that the magnetic field can be detected best when the magnet is oriented perpendicular to the front face of the phone (i.e., along the Z axis). In that orientation, we could detect the magnetic field at a maximum distance of 5 inches. However, at this distance, the magnetic field is not strong enough to send the trigger signal without bit errors. The field strength in this case hovers around the Earth's magnetic field, and fluctuates quite a bit.

Figure 3(b) depicts the maximum distances at which we were able to *send a signal* consistently without any bit errors. We were able to send a signal to the malware application at a maximum distance of 3.5 inches. At this distance, the magnetometer in the phone was able to detect a magnetic field of 60 microtesla from the electro-magnet, which is high enough to distinguish itself from the background/natural magnetic field.

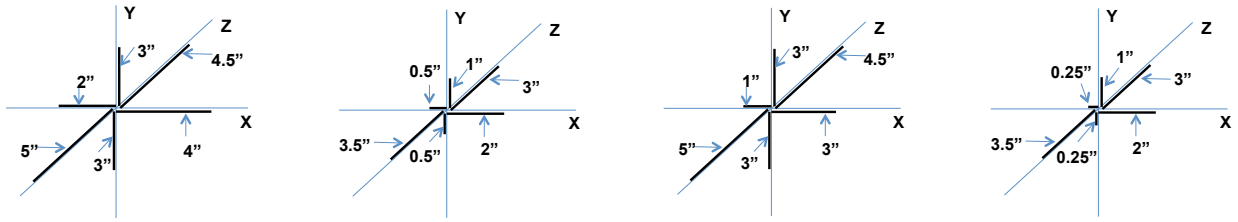
We next repeated the above experiment by covering the electro-magnet with plastic. As depicted in Figures 3(c) and 3(d), the magnetic field is, however, virtually unchanged even in this case.

These results confirm that the magnetic channel is feasible in many scenarios where the magnet can be placed close to the phone without worrying about shielding effects.

5.6 Power Consumption

Finally, we investigated the power consumption of the sensors. We tested the power consumption by keeping the sensors running constantly for 10 minutes. We found that, in practice, sensors use a small amount of power. For example, the magnetometer, accelerometer, and light sensors cause virtually no change in battery consumption. Even the microphone consumed less than 1% of the battery capacity over regular consumption rate during a 10 minute period. Further details of sensor power usage can be found in Table 5.

The power consumption analysis shows that the malware application, even when accessing the sensors, does not cause significant power drain in practice. We point out that only the steganographic



(a) Without cover (detection) (b) Without cover (signalling) (c) With cover (detection) (d) With cover (signalling)

Figure 3: Magnetic field distance tests. (a) and (c) show the test where presence of a magnetic field was detected by the malware. (b) and (d) show the test where we successfully sent the trigger to the malware with no bit errors. See Figure 2 for reference axes.

audio channel requires potentially expensive Fast Frequency Transform (FFT). The other channels do not use FFT. And while FFT does require computation resources, prior research shows it is feasible to reduce energy consumption [7]. Optimizations can parallelize FFT computation on ARM processors which are widely used in smartphones [35]. Another mechanism for evading power drain analysis is that, the attacker may send triggers only at certain times(e.g., once-a-day during telecast of a TV program where the trigger signal has been embedded). This significantly reduces likelihood of detection while still giving the attacker a periodic trigger window.

Sensor	**Power consumption in 10 minutes** (% of battery capacity)
No sensor	< 1
Microphone	2
Light sensor	1
Magnetometer	< 1
Accelerometer	< 1

Table 5: Power consumption of sensors.

6. DISCUSSION

6.1 Summary of Results and Further Analysis

Our experiments demonstrate the feasibility of sending command and control trigger messages to smart phones and similar devices over out-of-band covert channels.

Range: The direct audio channel exhibited the longest range – more than 55 feet indoors, 45 feet outdoors. It can also tolerate real-life background noise. We were able to achieve these communication ranges using only low-end PC speakers with minimal amplification and low-volume. In practice, we envision the audio channel to use better quality speakers or sound systems present in Televisions or other multimedia systems. The direct audio channel also has the advantage of working both indoors and outdoors. Our experiments also demonstrated that the audio reception works even if the phone will be kept in the user's pocket or a purse. However, we do not expect it to work when the phone is stowed inside leather pouches or inside other thick material. Therefore, our experiments provide evidence that the audio based command and control via TV or radio programs, background music services, Internet TV program and musical greeting cards is feasible. This is also true for our steganographic channels, although our current experiments with these channels were conducted in a typical office setting.

The ambient light channel has shorter range compared to the audio. However, it has the advantage of sending a trigger message simultaneously to all the phones in a targeted building. The light channel works best at night or in places with low illumination. During daytime, the presence of sunlight introduces a large amount of ambient light (a form of background noise), due to which receiving the messages over long distances might be difficult. However, when using large screen televisions, the messages can be relayed over reasonably long distances.

The magnetic channel, as expected, has the shortest range. This is due to the inherent property of magnetism – magnetic signals rapidly fade as they travel in the air. However, even this short range channel can be surreptitiously exploited in scenarios where the signal transmitter is hidden in the doorframe or on an elevator. The magnetic channel clearly would work in most, if not all, scenarios irrespective of the way the phones are stowed or carried by users.

Channel Bandwidth: A limitation of the sensor-based channels is their low-bandwidth. In our current implementation, we were able to achieve a maximum bandwidth of 5 bits per second with these channels. This limit is attributed to a number of different factors such as the limited sensitivity of mobile device sensors, the low sampling rate enforced by the Android OS and the need for undetectability of transmission. However, since command and control messages – especially the triggers – need not be too large, we believe that even this low bandwidth will be sufficient in most scenarios. To offset the low transmission rate, the messages can still be transmitted over large time spans, such as over an entire TV program or multiple occurrences of the same program. One possibility to improve the bandwidth is to employ multiple channels simultaneously. For example, the messages may be delivered in parallel over the audio channel as well a light channel, say while watching a TV. We believe the bandwidth limit to be inherent in the design of the sensing-enabled channels. This represents the obvious trade-off in using these channels for undetectable command and control when compared to other wired/wireless channels.

6.2 Possible Defense Mechanisms

Although it may be difficult to detect and prevent the out-of-band command and control presented in this paper, we recognize this to be an important problem worthy of further investigation. To this end, we suggest a few preliminary set of defenses.

An intrusion detection application running on the mobile phone itself can detect the sensor based signaling and prevent the malware application from receiving it by monitoring the sensor data stream. To this end, we envision two approaches. First, instead of giving the applications direct access to the sensors, a virtualization layer can be created between the sensors and the applications. The virtualization layer includes a monitor that constantly analyzes the sensor accesses by different applications as well as the sensor data streams to determine whether any malicious activity is taking place. The virtual sensor monitor may use machine learning techniques to determine the natural behavior of benign applications when run by a given user. The downside of this approach is that it may be rather heavyweight and requires the phone to monitor a large number of sensor readings. Second, in order to determine whether a malicious signal is present in the sensor readings, the phone can take random sensors samples and analyze them.

Another phone based defense could be based on the level of power consumed by applications that access sensors. A malicious application listening on to one or more of the sensors is likely to consume slightly more power compared to other applications (see Table 5). The feasibility of such a defense against malware is demonstrated in [23]. A related approach using memory footprints [21] can also be used to detect such stealthy mobile malware applications.

Finally, another line of defense is to monitor external media streams for malicious communication. For example, media streams such as radio, music, or video streams can be monitored by service providers prior to the transmission. These mechanisms can be effective against some of the attack scenarios presented in this work such as the use of TV or radio programs. However, it may be very difficult to detect the presence of our steganographic channels.

7. RELATED WORK

Most closely related to our work is the Bluetooth based command and control architecture proposed in [46]. In this approach, Bluetooth is used as the primary means of communication between the botmaster and infected devices. In particular, a hybrid approach is suggested whereby a set of seed nodes are first selected with which the botmaster communicates over the traditional network (such as SMS), and these seed nodes then relay the messages to other nodes in close physical proximity over Bluetooth. Based on public Bluetooth traces, [46] demonstrated that such an infrastructure of command and control is feasible. However, it exhibits high latencies in propagating messages. More specifically, such messages can reach about 70% of infected devices within 24 hours. Although some of our channels also exhibit similar latencies, many other channels offer an almost real-time delivery guarantees to an extremely large number of devices. Moreover, many of our channels do not at all rely upon a traditional infrastructure (such as SMS) and provide a means of undetectable communication, and some (such as audio) even possess communication range higher than that of Bluetooth.

HumaNet [8] is a similar architecture to that proposed in [46]. In contrast to [46], however, in this approach, the command and control messages are delivered only via phone to phone ad hoc communication (such as Bluetooth or WiFi ad hoc mode). This work also demonstrated the latencies similar to that of [46]. Compared to this work, the focus of our paper is on out-of-band communication of different flavors, including phone-to-phone communication in some cases.

Another related work is SkyNET [42] which makes use of compromised personal networks (such as personal WiFi) as a platform for command and control message dissemination. Other mobile botnets have been reported in scientific publications that make use of other services available on mobile devices for communication such as SMS/MMS [13, 19, 37, 49, 50] and available open WiFi networks [26]. Our work, on the other hand, utilizes out-of-band communication channels for command and control which are much harder to detect compared to the use of cellular or WiFi services.

The explicit use of steganography for the purpose of undetectable command and control, as employed in our stego audio channel, is also inline with Stegobot [39]. This is a botnet architecture that uses social networking based steganographic channels for command and control. In particular, image steganography is used to achieve this goal. Our steganographic audio channel, in contrast, embeds covert messages which are transmitted in real-time *over the air*, and not into static files or images.

Out-of-band communication in general has been employed in many security applications before. These include: proximity-based secure association of personal wireless devices (see surveys and usability studies [20, 22, 27, 28]) and user authentication [43]. Besides identifying a new application of out-of-band communication, i.e., undetected command and control, we have proposed many novel channels not previously explored.

8. CONCLUSION

The threat of mobile malware is rising rapidly. According to a recent report by Lookout Security [3], the Android phone users are becoming more and more vulnerable to malware. The same report states that, in several cases, legitimate applications were repackaged with malware code and distributed via the official Android app store or via updates/advertisements. Android and iPhone botnets have already been proposed [37, 49]; some have even appeared in the wild [49]. However, most of these mobile botnets used SMS or the Internet to build the command and control infrastructure – similar to traditional botnets, and are therefore detectable via traditional botnet defense mechanisms.

In this paper, we investigated the feasibility of sensing-enabled covert channels in mobile phones. Malware using such channels will be very difficult or impossible to detect using traditional means, because such the underlying command and control channels exploit non-network air-gaps to communicate. Our proof-of-concept prototype exemplifies this emerging problem – using off-the-shelf hardware and popular Android-based mobile phones, we were able to send surreptitious command and control messages without using any wireless or cellular networks. Our prototype malware application received the messages embedded in music, video, household lighting, or magnetic fields.

Malware with the capability of using such sensor-based covert channels can also open up new threats such as the creation of localized botnets and geo-targeted attacks, which we explored briefly in the paper.

Acknowledgment

This research was supported by a 2012 Google Faculty Research Award, the Office of Naval Research Grant #N000141210217, the Department of Homeland Security Grant #FA8750-12-2-0254, and the National Science Foundation under Grants #0937060, #1201927, and #1228236.

9. REFERENCES

[1] ButtKickers – Low Frequency Audio Transduceers. Available at `http://www.thebuttkicker.com/`.

[2] iPhone Microhone Frequency Response. Available at `http://bit.ly/IDWfei/`.

[3] Mobile Threat Report. Available at `http://bit.ly/ImefoN/`.

[4] PPLive. Available at `http://www.synacast.com/en/`.

[5] SopCast - Free P2P internet TV. Available at `http://www.sopcast.org/`.

[6] Suggested Viewing Distance for HDTV (Calculator). Available at `http://hdinstallers.com/calculator.htm`.

[7] M. Aboleaze and A. Elnaggar. Reducing memory references for FFT calculation. In *Proc. of the International Conference on Computer Design*, pages 26–28, 2006.

[8] A. J. Aviv, M. Sherr, M. Blaze, and J. M. Smith. Evading cellular data monitoring with human movement networks. In *Proc. of USENIX HotSec*, 2010.

[9] H. Bassen. Radiofrequency interference with medical devices. a technical information statement. *IEEE Engineering in Medicine and Biology*, 17(3):111–114, 1998.

[10] L. Cai and H. Chen. Touchlogger: inferring keystrokes on touch screen from smartphone motion. In *Proc. of USENIX HotSec*, 2011.

[11] E. Cochran, J. Lawrence, and C. Christensen. Quake-Catcher Network. Available at `http://qcn.stanford.edu/`, 2008.

[12] P. Dhungel, X. Hei, K. W. Ross, and N. Saxena. The pollution attack in p2p live video streaming: measurement results and defenses. In *Proc. of P2P-TV*, 2007.
[13] G. Geng, G. Xu, M. Zhang, Y. Yang, and G. Yang. An improved sms based heterogeneous mobile botnet model. In *Proc. of IEEE ICIA*, 2011.
[14] K. Gopalan and S. Wenndt. Audio steganography for covert data transmissions by imperceptible tone insertion. In *Communication systems and applications*, Available at `http://qcn.stanford.edu/`, 2004.
[15] D. J. Griffiths. *Introduction to Electrodynamics (Third Edition)*. Prentice Hall, 1999.
[16] J. Han, E. Owusu, T.-L. Nguyen, A. Perrig, and J. Zhang. ACComplice: Location Inference using Accelerometers on Smartphones. In *Proc. of COMSNETS*, Jan. 2012.
[17] Harris Interactive. Teenagers: A Generation Unplugged. Available at `http://bit.ly/IyH71E`, 2008.
[18] Harris Interactive. The Harris Poll - Cell Phone Usage Continues to Increase. Available at `http://bit.ly/IpgefF`, 2008.
[19] J. Hua and K. Sakurai. A sms-based mobile botnet using flooding algorithm. In *Proc. of WISTP*, 2011.
[20] I. Ion, M. Langheinrich, P. Kumaraguru, and S. Capkun. Influence of user perception, security needs, and social factors on device pairing method choices. In *Proc. of SOUPS*, 2010.
[21] M. Jakobsson and K. Johansson. Retroactive detection of malware with applications to mobile platforms. In *Proc. of USENIX HotSec*, 2010.
[22] R. Kainda, I. Flechais, and A. W. Roscoe. Two heads are better than one: Security and usability of device associations in group scenarios. In *Proc. of SOUPS*, pages 1–13, 2010.
[23] H. Kim, J. Smith, and K. G. Shin. Detecting energy-greedy anomalies and mobile malware variants. In *Proc. of MobiSys*. ACM, 2008.
[24] R. Kim. The World's a Cell-phone Stage. SFC, Available at `http://bit.ly/IpgSdf`, 2006.
[25] Knowledge Networks. New Study Shows Mobile Phones Merging New, Established Roles: Communicator, Shopping Aide, Entertainment and Research Hub. Available at `http://bit.ly/IDWyWv`, 2008.
[26] M. Knysz, X. Hu, Y. Zeng, and K. G. Shin. Can open wifi networks be lethal weapons for botnets? In *Proc. of INFOCOM*, to appear, 2012.
[27] A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang. Serial hook-ups: A comparative usability study of secure device pairing methods. In *Proc. of SOUPS*, 2009.
[28] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. Caveat emptor: A comparative study of secure device pairing methods. In *Proc. of PerCom*, 2009.
[29] D. Lee. College Student's Hand-phone Usage Culture Survey. In *University Culture Newspaper*, 2002.
[30] S. Lee, K. Fu, T. Kohno, B. Ransford, and W. Maisel. Clinically significant magnetic interference of implanted cardiac devices by portable headphones. *Heart rhythm : the official journal of the Heart Rhythm Society*, 6(10), October 2009.
[31] S. Lohr. As Cellphones Bulk Up, How Much Is Too Much? Available at `http://nyti.ms/IEk3hV`, 2005.
[32] C. V. Lopes and P. Q. Aguiar. Acoustic modems for ubiquitous computing. *IEEE Pervasive Computing, Mobile and Ubiquitous Systems*, 2(3):62–71, July-September 2003.
[33] Market Analysis and Consumer Research Organisation. Study of Mobile Phone Usage Among the Teenagers And Youth In Mumbai. Available at `http://bit.ly/JrjaCQ`, 2004.
[34] P. Marquardt, A. Verma, H. Carter, and P. Traynor. (sp)iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proc. of ACM CCS*, 2011.
[35] P. Meerwald. KissFFT and ARM NEON. Online at `http://bit.ly/WtSjDI`, 2011.
[36] E. Mills. Attacking home automation networks over power lines. CNET, Available at `http://cnet.co/JCm8Ji`.
[37] C. Mulliner and J.-P. Seifert. Rise of the iBots: Owning a telco network. In *Proc. of MALWARE*, Oct. 2010.
[38] Muzak Inc. Music and More for any Businesses. Available at `http://www.muzak.com/`.
[39] S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, and N. Borisov. Stegobot: a covert social network botnet. In *Proc. of IH*, 2011.
[40] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang. ACCessory: Keystroke Inference using Accelerometers on Smartphones. In *Proc. of HotMobile)*, Feb. 2012.
[41] W. K. Park. Mobile Phone Addiction. In *Mobile Communications: Re-negotiation of the Social Sphere*, 2006.
[42] T. Reed, J. Geis, and S. Dietrich. Skynet: a 3g-enabled mobile attack drone and stealth botmaster. In *Proc. of USENIX WOOT*, 2011.
[43] N. Saxena and J. H. Watt. Authentication technologies for the blind or visually impaired. In *Proc. of USENIX HotSec*, 2009.
[44] R. Schlegel, K. Zhang, X. yong Zhou, M. Intwala, A. Kapadia, and X. Wang. Soundcomber: A stealthy and context-aware sound trojan for smartphones. In *Proc. of NDSS*, 2011.
[45] A. N. Selian. Mobile Phones and Youth: A Look at the U.S. Student Market. Available at `http://bit.ly/ZnVMA4`, 2004.
[46] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee. Evaluating bluetooth as a medium for botnet command and control. In *Proc. of DIMVA*, 2010.
[47] P. L. Sunderland. Fieldwork and the Phone. 1999.
[48] T. A. Wikle. America's Cellular Telephone Obsession: New Geographies of Personal Communication. In *Journal of American and Comparative Cultures*, 2001.
[49] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi, and Z. Tianning. Andbot: towards advanced mobile botnets. In *Proc. of USENIX LEET*, 2011.
[50] Y. Zeng, K. Shin, and X. Hu. Design of SMS commanded and controlled and P2P structured mobile botnets. In *Proc. of ACM WiSec*, 2012.

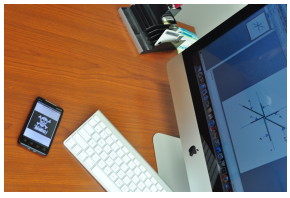
Appendix

(a) We used low-end PC speakers to transmit the audio signal.

(b) A malware-infected phone can detect the trigger signal embedded in audio.

Figure 4: Equipment set-up for the audio channel

(a) A user is walking past the door with his phone inside the pocket, yet within close range of the transmitter.



(b) Phone placed flat on the desk, receiving signal from the overhead light.

Figure 5: Equipment set-up for the magnetic and light channel

For more images of our experimental setup, please visit `http://secret.cis.uab.edu`