Evaluating the threat of epidemic mobile malware

Christian Szongott, Benjamin Henne, Matthew Smith
Distributed Computing & Security Group
Gottfried Wilhelm Leibniz University of Hannover
Email: {szongott,henne,smith}@dcsec.uni-hannover.de

***Abstract*—While mobile malware has played a relatively small role compared to the behemoth of desktop malware, the changes both in the capability as well as in the proliferation of the mobile devices will steadily increase the attractiveness of mobile devices as resources to be attacked. The increased usage and connectivity of mobile devices opens up a much larger set of attack vectors to compromise these devices. In this paper, we discuss how the new features of mobile devices opens up the capability to create a new generation of mobile malware which is capable of realistically spreading epidemically on a fully mobile infection vector. We present a prototype of such a mobile malware that uses these features to replicate itself and spread over a mobile device-to-device vector. Using simulations we give quantitative support for the number of mobile devices and conditions needed to for an epidemic spread of mobile malware and present infection scenarios in downtown Chicago. Our results show that the recent growth and market dominance of just a handful of mobile phone companies and the trend towards mobile operating systems monoculture has already created a viable substrate for epidemic mobile malware.**

I. Introduction

In the past malware attacks only affected desktop PCs and servers while mobile devices like phones and tablets have been ignored by attackers. There are different reasons that led to this deceiving situation pretending mobile users to be safe from such attacks.

The main driving force behind any malware is the cost-benefit ratio. The big mainstays of malware Spam and DDoS attacks were/are less well suited to mobile devices, leaving mainly direct monetization oriented attacks such as spyeye's attempt to steal mobile banking m-tans or malware which calls premium numbers. Also, the target environment was very fragmented with a large number of different devices and operating systems. This meant that to reach a significant number of targets, exploits in many different systems needed to be found and attack and payload code needed to be crafted for each system accordingly. Even today's spyeye trojan relies on the PC-based trojan to locate and infect the mobile devices and only works in conjunction with the PC trojan. This dampens the traditional economy of scale approach of monetizing malware, since it is very difficult and costly to create a large infection base. These factors have led to a rather poor cost to benefit ratio in the past.

However, the increased capabilities and usage scenarios make mobile devices a more lucrative target nowadays. Firstly mobile devices had very limited capabilities in the past compared to today's smartphones. Only GSM-based and Bluetooth-based attacks were possible. Today smartphones contain multiple communication interfaces, like WiFi, Bluetooth etc. The use of Near Field Communication (NFC) technology to enable wireless payment will also have impact on the worth of mobile phones as a target for mobile malware. Secondly, the increased usage and connectivity of mobile devices opens up a much larger set of attack vectors to compromise these devices. While a couple of years ago Internet usage on a mobile device was slow and painful and practiced only by a few hardcore users, it is now mainstream, with users regularly checking email, surfing the Internet, using social networks and a whole host of other Internet enabled services.

Apart from the increased usage and capabilities of mobile phones, the consolidation of the mobile operating system market is also an important factor to consider. While only a couple of years ago each mobile phone company had its own mobile phone operating system (often even several) the current trend in the mobile operating system market seems to suggest that we will soon be left with only two or three different smart device operating systems: Android, iOS and maybe one from Microsoft or Nokia. This significantly facilitates the development and deployment of mobile malware, since a much larger user base can be reached with a single exploit and payload code. While all these changes significantly increase the potential for mobile malware, there is still one component missing to create an epidemic mobile malware outbreak and that is a device-to-device infection vector. Felt et al. showed in their survey [1] that malware exists for all of today's mobile operating systems but none of them being capable of replicating itself on unmodified devices or without active involvement of the user.

To show the dangers that come along with new features of modern smartphones we have implemented the key components of a self-replicating proof-of-concept malware for the iOS platform. We achieved this by modifying an existing jailbreak mechanism and are able to install arbitrary software during the jailbreak process. Our proof-of-concept malware utilizes this to install webserver and hotspot software to start an evil twin attack. Usability features like the automatic reconnection for known WiFi access points and the support for captive portals enable our malware to spread to other mobile devices nearby.

Using appropriate simulations we give quantitative support for the number and conditions needed to get a large-scale infection base and show results based on the downtown area of Chicago. These results show under which circumstances an

978-1-4673-1430-5/12/$31.00 ©2012 IEEE

epidemic spread of a mobile malware is possible and what parameters influence the epidemic character of the spreading.

This paper is organized as follows. In Section 2 we will give an overview of existing simulation works in the field of epidemiology models and show their drawbacks that lead to unrealistic results. In Section 3 we briefly describe our proof of concept malware and show how it works. In Section 4 we describe our simulation environment, explain which assumptions were made and how the logic of different agents and their interactions have been implemented. In Section 5 we compare simulation models of other simulators to ours and show how important it is for analysis like this to take details such as movement patterns and device usage patterns into account. In section 6 we present further results of our simulations and show the potential of mobile malware attacks today and in the future. Section 7 concludes the paper and gives an outlook to possible future work in this field.

II. Related Work

Numerous models for the simulated study of mobile malware and possible countermeasures have been proposed in literature. Many use pure mathematical models often based on the mathematical epidemiology models of natural viruses [2]–[7], while others use mobile agent-based simulations [8], [9]. A particular drawback of mathematical models such as the susceptible-infectious model [10] is that these models only consider temporal dynamics of an infection, but no spatial dynamics i.e. the spreading of an infection in space or highly viral locations. While some agent-based simulators also model a spatial component, most of these simulations are currently quite simple using very basic assumptions like the mathematical models (e.g. homogenous users, random walk models in empty terrain and instantaneous infection). One notable exception is the Siafu simulator [11] which uses a gradient image of a street map to allow the agents to move along real streets.

Mascetti et al. [12] show the impact of user movement on the evaluation of mobile malware. They compared experimental results of evaluations with mostly random movement of users on the one hand and generated movement data created with the Siafu [11] simulator on the other. They show there is already a measurable difference between the simple random walk models used by most malware infection models and the street-based random walk model of Siafu. Other work [13] uses location traces from mobile phone carriers as a basis for their research. These kinds of traces cannot be used to explore the spread of a mobile malware since radio cell based location information is too coarse-grained for the infection scenario. To be able to simulate the spreading of a malware that relies on vicinity we need to be able to model user movement through time and space with accuracy of meters and seconds.

UDel Models is a simulation suite for simulating MANETs and urban mesh networks with the focus on realistic mobility and propagation. The 3d mobility model includes pedestrian movement in multi-story buildings and on outdoor sidewalks. In [14] Kim et al. present a layered mobility model that consists of activities and sub-tasks based on time use and management research studies. Node dynamics such as grouping and speed-distance relationships are supported as well. The mobility model is used to generate traces, which can be used with other simulators.

A sign for the danger of large-scale mobile infection bases is shown by Husted et al. [15]. They describe how malnets can be used to track the location of non-infected mobile devices and that a small percentage of oblivious detectors can track a device a significant amount of the time. Traynor et al. [16] show how a relatively small amount of infected mobile phones operating in a malnet can attack the mobile core network and cause significant network outages in the covered area.

In the field of mobile malware and especially malware for iOS devices some worms could be found in the past. Most of them are based on a jailbroken iOS device in combination with a standard root password. The iKee worm [17] was first observed in 2009. Since iKee only changed the background wallpaper, subsequent versions like iKee.b/Duh [18] have connected to a botnet control server downloading additional components and sent bank information from SMS messages to it. Antoher malware that runs on the device itself is the Siri Privacy Exposer (SPE) [19]. It runs a MITM attack by DNS spoofing and submits private information to attackers that are transfered to Apple's Siri servers. The iPhone/Privacy.A malware runs on a laptop, [20] scans the local network for attackable devices and steals personal information, like calendars, addressbooks, sms messages, etc. from them. All these malwares rely on the fact that the iOS devices are jailbroken and have an unchanged root password. Damopoulos et al. have presented an iPhone Stealth Airborne Malware (iSAM) [21], that is able to run a variety of attacks on the infected devices. In contrast to our approach the propagation of the malware is realized by sending a large number of SMS messages that lead the victim to a prepared PDF file containing the exploit code. Thereby this attack depends on a working internet connection and the propagation mechanism does not take spatial proximity into account.

III. Mobile Malware

In this section we will describe the components that are necessary for our epidemic malware. The malware is based on an extension to the "evil twin" attack [22]. In the evil twin attack an access point (AP) inpersonates a known SSID in the hope that a victim connects to the rouge AP. This type of attack has been discussed in literature before [22]–[24], however it did not receive much attention since it was only discussed in the context of infrastructure-based Wi-Fis, where targeted attacks against specific locations were the goal. While the potential effect of the evil twin attack is impressive, their real world applicability at the time was limited due to the difficulty of setting up the evil twin and the targeted nature of the attack. In this paper we introduce an adaptation of the evil twin attack leveraging the changed landscape of mobile devices to create a far more potent attack type, which we dub the Mobile Evil Twin (MET) Attack. Unlike the evil twin attack the MET does not aim at subverting a single AP to sniff out users credentials

or execute MITM attacks, but to misuse the weaknesses and the usability features of mobile wireless devices to create an epidemic mobile malware infection substrate.

A. Implementation

In the following, we will describe the key components of the MET malware and the steps executed in the mobile spreading of the malware. All components were separately tested in the lab using iOS 4.3.3 running on an iPhone 4, an iPad 1 and an iPad 2.

Step 1: Evil Twin hotspot: The first step of the attack is the exploitation of the auto-reconnection feature to known hotspots. If a mobile device attempts to access the Internet, for instance to browse the web, send or receive email or an App makes a connection, the device checks to see if there is a wireless network available before using 3G. The mobile malware attack starts with the initial infection host broadcasting an SSID corresponding to a well used SSID such as those used by public hotspots of cafes, restaurants, universities or telephone companies. If a mobile device has previously been connected to a legitimate AP that SSID is stored in a list of known networks to which the device will connect again if the SSIDs match. Since SSIDs are not authenticated our rouge device can spoof an arbitrary SSID. Once the mobile device is connected all Internet activity of the device goes via our hotspot. With this setup, we can run a rogue Wi-Fi hotspot modifying unencrypted communication sent through it and thereby injecting malicious content. A usability feature of iOS devices is

Step 2: Exploitation: Once the mobile device is connected to the MET hotspot our proof of concept malware uses `pf`, the iOS internal firewall and packet forwarding engine, to redirect all incoming traffic on port 80 to the locally deployed `lighttpd` web server where the malicious payload is hosted. Since public hotspots often use captive portals to request user credentials or payment information there is an inbuilt mechanisms in iOS to display a browser popup windows as soon as the device connects to an AP. We use lighttttpd web server to then serve a browser based exploit into this popup. We extended the jailbreakme.com exploit for this purpose. The exploitation jailbreaks the device and transfers the MET to the freshly broken device. Using an exploit like the jailbreakme.com exploit is just one example.

Step 3: Preparing a new Evil Twin: Once a kernel-level exploit was executed on a mobile device the attacker gains complete control over that device, including the ability to control network and radio functions. Our propagation concept makes use of the ”personal hotspot” feature (Apple iOS $\geq$4.3). This hotspot functionality can be used by the device owner to share their 3G Internet connection with others by creating a mobile Wi-Fi access point. It conveniently bundles a small DHCP server, routing capabilities and Wi-Fi channel management and the setup is trivially easy. During our mobile malware attack the mobile hotspot gets activated using the target SSID making the device infectious as well.

By deploying *lighttpd* as a lightweight web server on the infected mobile phone and rerouting all HTTP traffic of connected devices to it, we can deliver malicious code within any HTTP response and thus infect more mobile devices. Now, the infection cycle starts back at step 1.

B. Evaluation

We implemented the key components of the MET malware and tested them in a laboratory environment to evaluate the time needed to infect a new device and the battery drain created by running the MET, since both these factors play an important role for the epidemic qualities of this kind of malware.

The jailbreakme.com exploit and all the required libraries and code that comprise MET are 10 MB in size. We conducted tests to discover how long two devices would need to be in range of each other to transfer MET. Figure 1 shows that the download time at the range of 1-2 meters was about 9 seconds. At distances of up to 15 meters download times rise to approximately 12 seconds, increasing to more than 32 seconds at a 25-meter distance. The measurements were conducted outside in a populated area with four other Wi-Fi hotspots active in the immediate vicinity. Each measurement was repeated five times and the standard deviation is shown in the figure.

The initial exploit takes less than one second. Since starting the required servers and configuring the packet forwarding engine takes approximately 1 second, these times can be neglected compared to the long times needed for the transmission of the files.

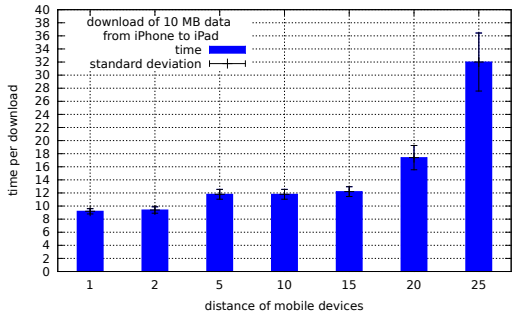


Fig. 1. Download times for the mobile malware files for different distances

Figure 2 shows the battery consumption of the MET measured over a six hour period. The red line shows the battery consumption of the uninfected phone when it is idle. The green line shows the battery consumption of an infected phone with our malware hotspot activated but otherwise idles. The blue line shows the battery consumption of a device running the malware, which infects another device every 20 minutes. We use this measurement to estimate the battery consumption per download. As can be seen the malware, if operated continuously, is a significant drain on the battery. However the current version of the malware has not been optimized for stealth or longevity.

IV. Simulation

Since large-scale malware infections cannot appropriately be analyzed due to legal and ethical aspects we chose to

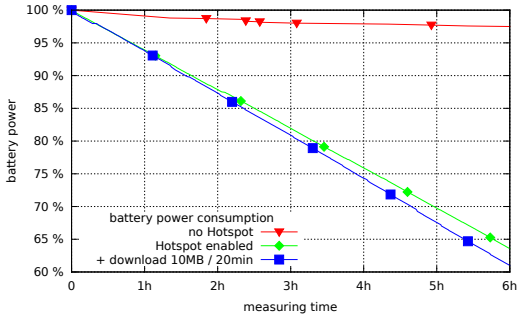


Fig. 2. Battery consumption with disabled/enabled hotspot software and with repeating transfer of malware files

simulate mobile device owners as well as the infection process itself. We simulate the spread of a mobile malware that is based on a device-to-device infection using our dedicated mobile security simulator [25] which models the relevant aspects for a study of epidemic spread of mobile malware. In particular, we postulate that user behavior, actions and social gathering locations are of relevance and have not been taken into account in mobile malware analysis for now. The mobile security simulator models a realistic environment and also includes logic for locations such as cafes, malls, etc. Additionally it allows for different user groups with different behaviors to be modeled. Besides personal context the simulated world has properties, which can influence the behavior of people and locations like cafes or malls can attract users creating longer exposure to high-density environments.

A. Environment and Assumptions

In the following we presume a mobile malware that is capable of infecting other devices by the use of faked free WiFi access points. A mobile device gets infected if it actively tries to access the Internet and stays in the vicinity of an infectious mobile device for a specified amount of time.

The initial infect is realized by placing a malicious mobile device at an haunted place. We chose a Starbucks cafe to be the initial infection location, since it meets the requirements for the first step of the infection process. People visiting this cafe get infected with a given probability if using a vulnerable mobile device.

1) Population: To get an idea of the possible spreading characteristics of such a malware, we try to estimate the relevant environmental parameters to configure the simulation. Of particular relevance to the study of the potential for an epidemic spread of the malware are the number of users who can potentially be affected and their device usage patterns. We chose downtown Chicago as the arena in which to study the malware. Using OpenStreetMap[1] data we modeled streets, public places and selected locations like cafes and hotspots.

The first question, which needed to be answered, was how many infectable devices are likely to be in the downtown area. Like Husted et al. [15] we use transport statistics to estimate roughly how many infectable devices to simulate. Including commuters the number of people using smartphones in downtown Chicago is roughly 400.000. While multi-platform exploits are of course possible our malware can only infect iOS. Thus we need to estimate how many iOS devices are likely to be used by the population. A recent comScore study [26] found the iOS market share to be 12% of the mobile phone market in the US. Only a small share of the remaining population is running an iOS version that is infectable through our malware. So we have a population of roughly 4,000 devices. However, since the trend to a monoculture of mobile operating systems continues and multi-platform are possible in theory we ran most of our simulations with a population of 10,000 people.

2) Battery consumption: Based on the lab test results shown in Figure 2, we estimated the battery drain bat_i from running our malware and apply this value once a device gets infected. To consider the battery drain from a running malware we apply a higher battery consumption level for infected devices within our simulation. We also subtract an additional amount of battery power per infection process to account for the power needed to transfer and manipulate a victim's device.

3) Infection duration: The infection duration is split up into the transmission and the set up time that is needed, before an infectable device gets infectious as well. Based on our measurements in Section III-B we will take conservative 15 seconds for the total infection time splitting up into 12 seconds pure transmission time and a further 3 seconds before the device becomes infectious.

4) User Actions: As mentioned above the simulated scenario takes place in downtown Chicago ("The Loop"), which is a part of Chicago that is dominated by commuters, which work, shop or stroll around on the streets, at public locations and in buildings. The high number of people rises the risk of an epidemic spread and lots of commuters carry out the infect when leaving the simulated area. We include a simple user model to approximate different user behaviors (i.e. shopping, sitting in a cafe, moving from location to location, walking or sitting in a park, etc.) as well as different user groups. This is necessary since an infection relies on some form of Internet activity to trigger an infection. In our scenario we simulate five different user groups and four different actions.

The possible actions are: a) *walking*: A person randomly chooses a destination on the map and walks to it. Once at the destination the person pauses for a certain time. b) *public space*: A person walks to a public space such as a park (defined in the geo-spatial data) and spends some time there. c) *location*: A person visits a location such as a cafe or a mall and spends an amount of time there. When performing this action a location is chosen from the set of locations defined in the geo-spatial data. This removes the person from the global geo-spatial simulation and transfers him to an independent mini-simulation of the location. d) *leave*: A person leaves the simulated area. This is an important feature since a closed environment has different characteristics to an open one. While in a closed system all devices will become infected over time, in an open system it is eventually possible that some devices

[1]OpenStreetMap project: http://www.openstreetmap.org/

enter and leave the infection area without getting infected. It also means that infected devices can leave the simulation and thus reduce the infection substrate making further infections less likely in the simulated area.

A state machine for each person of a user group decides which next action a person executes. Importantly the groups also define the Internet access habits of the group members (i.e. how often the phone is activated / how often the Internet is used). The initial user groups used to study the mobile malware spread are: *Power Users*, *Window Shoppers*, *Cafe Visitors*, *Average People* and *Strolling People*. Table I shows the parameter values that are used in our simulation scenario. The column v shows the velocity (in m/s) and p the percentage of people being in this group. The following six columns describe how long (in minutes) a person waits at a given destination. The interval boundaries *wait min* and *wait max* values (in minutes) limit the normal distributed waiting time when a person arrives at a destination. The timing values t_{o} and t_{loc} set how often a person's device gets activated. There are two separate intervals: t_loc inside locations such as cafes (where users tend to user their mobile device more often) and t_o in the open. The next four columns show the probabilities the finite-state machine uses to determine the next state of a person when he arrives at a destination. The last two columns show the battery lifetime (in hours) of not infected and infected mobiles.

B. Infection

The Mobile Security & Privacy Simulator is an agent-based simulator with different usage and mobility patterns to model different types of users. Users can enter buildings and spend time there. We used a study on mobile phone usage presented by Karlson et al. [27] to set realistic phone usage patterns. Since the study focused on information workers we used these number as the upper bound and set a number of lower usage times for the majority of the population as a conservative estimate. In our simulation we differentiate between two distinct infection environments: roads and locations. On roads and in open space such as parks, the infection relies on the user's location, device usage, distance between infected and non-infected devices and time spent within communication range.

Only if the device is activated by the usage model and thereby tries to access the Internet, an infection can take place, significantly reducing the infection risk compared to the traditional infection models, as we will show in the results section. Once connected, the infection only occurs if the devices remain in communication range within 15 meters for at least 15 seconds based on the time needed to send the exploit code to the vulnerable device and execute it.

If a user's mobile device gets infected it henceforth can infect other devices that have been connected to well known wireless networks and thereby raise the local infection probability. When leaving a location, the infection is carried out and can in turn infect other people on the road, in other locations or public places. Outside the cafe, infection mainly does not rely on probability like mathematical epidemiology models, but takes logical, local and temporal conditions of the infection, the infectious and healthy people into account.

While locations such as cafes are contained in the OpenStreetMap data, their dimensions are not modeled as geospatial data. Thus, it is currently not possible to model movement and location within buildings. We utilize a mathematical model in these cases. To make up for the lack of detail in OpenStreetMap data we specify the base area and number of stories a location has manually. In the current setup the size of locations varies between 30 and 300 square meters. The infection probability for devices located in these locations is defined as follows: The area per infected visitor A_i is determined by dividing total area of the location as determined by its story count l and its base area a by the number of infected number of devices i.

$$A_i = \beta \cdot \frac{l \cdot a}{i} \tag{1}$$

The factor β is used to dampen the infection rate to account for the fact that the people in the building will seldom be distributed equally and to account for the fact that in buildings with many different infrastructure and ad-hoc networks they will disturb each other and thereby making an infection less likely as in the open. For this paper we set $\beta = 0.16$ based on experimental trials with our proof of concept prototype.

Due to competing signals we also lower the effective communication range to 5m. Based on the Wi-Fi range $r_{\text{Wi-Fi}}$ and A_i the spatial infection probability is calculated. Based on the device activation interval t_{loc} and the duration of his visit t_{visit}, the temporal infection probability is calculated. The combination of these two factors results in the final infection probability P_i.

$$P_i = \frac{t_{\text{visit}}}{t_{\text{loc}}} \cdot \frac{2\pi r_{\text{Wi-Fi}}}{A_i}. \tag{2}$$

V. Comparison of World Models

Simulation has often been used to analyze the spread of mobile malware (eg. [2]–[7] [8], [9]). However these studies usually rely on mathematical models or simple agent based simulations. To compare our simulator model with these existing models we created four distinct simulation world models to study the spread of mobile malware.

R-Z (Random Model, Zombie Infection): The first world model uses a combination of a simple random walk model

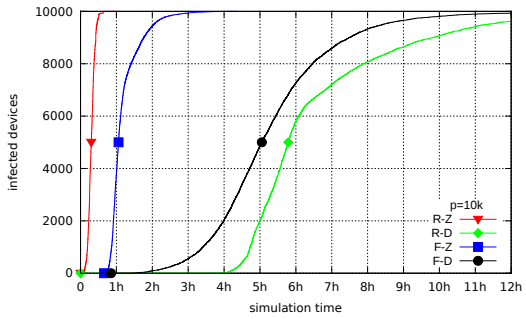


Fig. 3. Comparison of the four world models

TABLE I
SCENARIO VALUES FOR THE SIMULATION

type	v	p	**wait min**	**wait max**	**cafe min**	**cafe max**	**pub min**	**pub max**	t_o	t_{loc}	P_{walk}	P_{loc}	P_{pub}	P_{leave}	b	b_i
Power Users	1,5	20	1	30	30	120	5	30	15	15	70	20	6	4	10	6
Window Shoppers	1,2	30	1	30	5	20	2	30	30	20	40	18	40	2	18	8
Cafe Visitors	1,1	10	2	10	30	120	30	240	30	20	8	50	40	2	18	8
Average People	1,2	20	1	10	5	120	5	120	30	30	32	32	32	4	36	10
Strolling People	1,0	20	1	20	20	60	20	60	30	30	58	20	20	2	36	10

and zombie like infection, i.e. as soon as a non-infected device enters the range of an infected device infection takes place. This is an approximation of the random walk category of simulators, which do not take the constraint on infection imposed by real devices into account; *R-D (Random Model, Realistic Device Infection)*: The second world model combines the random walk model with our proposed realistic model of user and device behavior for the infection process, i.e. only if a device is used and is in range of an infected device for a defined amount of time an infection occurs. *F-Z (Full Model, Zombie Infection)*: The third world model combines our proposed realistic movement model and locations with the zombie like infection of scenario one. *F-D (Full Model, Realistic Device Infection)*: Finally, the fourth world model combines all our proposed improvements to the simulated study of mobile malware, i.e. realistic movement model, locations and device usage patterns.

First we present in Figure 3 the differences of the four world models, simulating a 12h period with a fixed number of 10,000 devices. In the figures p stands for the number of people in the simulation, t_i for the average interval between Internet activities of the devices and the four world models are abbreviated as R-Z, R-D, F-Z and F-D.

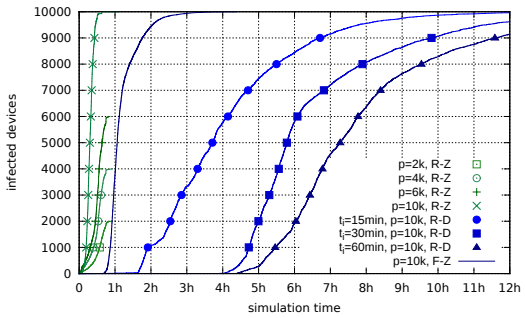


Fig. 4. Infection of 10,000 people using Random movement/Full Model and Zombie/Realistic Device Infections

Figure 4 shows world model R-Z (green) and F-Z (blue), i.e. the two world models with zombie like infection. As can be seen model R-Z achieves total infection in less than one hour. Even with only 6,000, 4,000 or 2,000 devices world model R-Z still achieves total infection in one hour. The addition of realistic movement patterns and locations in world model F-Z slows the infection slightly, however, full infection is still achieved in roughly two hours. Figure 4 also shows how important the addition of realistic device usage and infection behavior is. The green infection curve shows the infection rate of world model R-Z while the three blue curves show the infection rates with different Internet usage intervals of world model R-D. As can be clearly seen the addition of usage patterns slows the spread of the malware significantly extending the total infection time from under an hour up to almost ten hours.

VI. RESULTS

In the following we present the results of the simulation to give a rough idea of how the mobile malware could spread. Since the number of infectable devices has a significant effect on the epidemic spread of our malware, we conducted a parameter study over the number of devices in a closed world scenario, where no agent enters or leaves the simulated area. Figure 5 shows the normalized results of this study using our F-D world model. As can be seen from 5,000 devices onwards an almost total infection occurs within 12 hours. From 10,000 devices onwards almost total infection is achieved within 8 hours. Thus, the estimated 4,000 devices, we postulate to be currently infectable in today's downtown Chicago, are still just below the epidemic level which would see a total infection within one day.

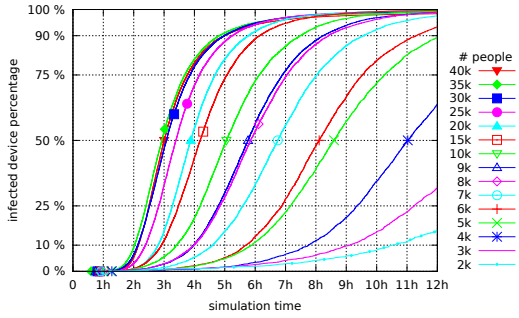


Fig. 5. Parametric study of infections using full model with realistic device infection

Due to the fact that the number of smartphones is steadily rising and an interested party could create a multi-device malware variant we conducted further studies on the epidemic qualities of a malware using 10,000 infectable devices. To show the effect of usage patterns on the spread of the mobile malware, Figure 6 plots the infection of the F-D model with different device activation intervals. As can be seen in the Figure the spreading of the malware has an epidemiological character for Internet usage intervals less than 30 minutes. Device usage intervals greater than 30 minutes lead to a

significantly slower but nevertheless substantial spreading of the malware.

Figure 7 shows the effect of moving from a closed system to an open system. In this simulation the commuters start leaving the downtown area after roughly 6-7 hours. As can be seen this has a significant effect on the infection rates quickly bringing the epidemic spread to a halt. However the commuters that left the simulated area are also spreading the infection outside of the downtown area.

Figure 8 shows a parameter study with different distributions of initial charge. Each simulation starts with a Gaussian distribution of initial charges in the range shown in the key of the figure. As can be seen the distribution of initial battery levels has a greater effect than the battery power drain. Surprisingly, even when a large number of devices run out of battery during the simulation there are still enough active infected devices to create almost full infection.

Finally Figure 9 shows an infection heat map which visualizes the spatial component of the infection process. Subfigure 9b shows the full model in which locations are included in the simulation and subfigure 9a shows a simulation where people do not enter buildings and only roads are included in the simulation. One can see that the addition of buildings and locations significantly changes the spatial component of infections and creates hot spots with a high infection probability.

To summarize, while simulation can only help estimate the true threat of an epidemic outbreak of mobile malware, we believe there are some key findings which can draw out of the simulations. Using device to device propagation mechanisms it is possible to achieve an epidemic spread of malware if the number of infectable devices is high enough. While the infection rate will vary depending on the exploit and the population, the infection rate can potentially be higher than with a traditional webpage- or AppStore-based approach which would require an action by the user.

VII. Conclusion and Outlook

In this paper we showed how mobile malware can spread epidemically on a device-to-device infection vector and almost entirely infect a metropolitan area such as downtown Chicago within a couple of hours. Although such attacks are possible

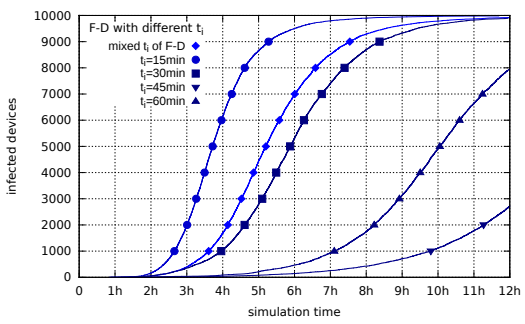


Fig. 6. Full model with realistic device infection using different Internet usage intervals

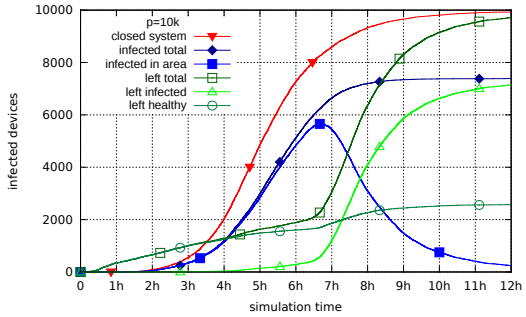


Fig. 7. Closed system vs. open system

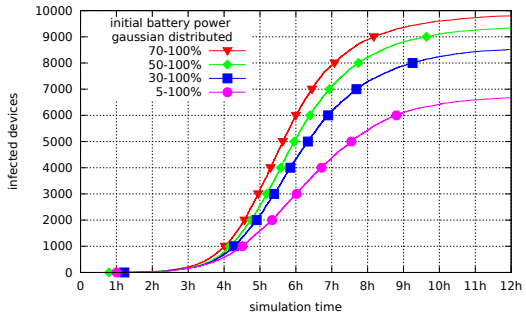
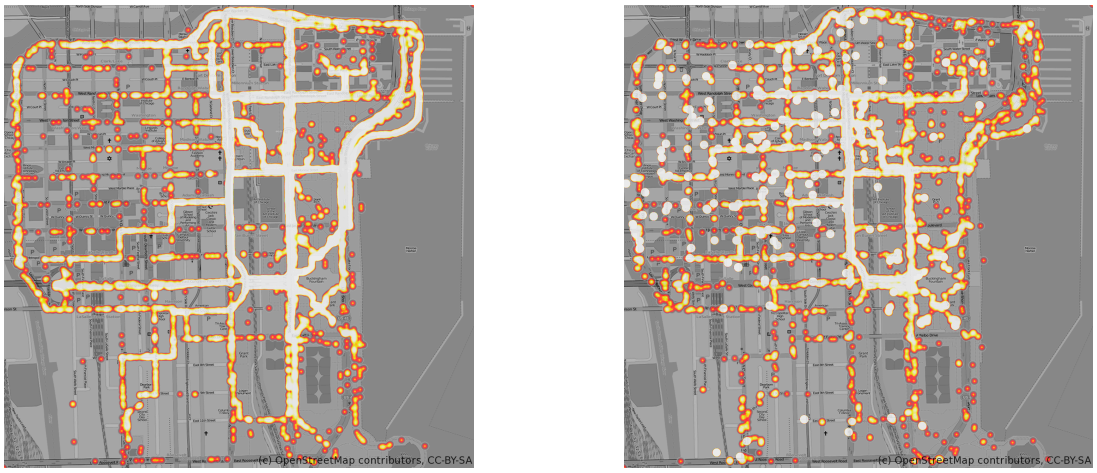


Fig. 8. Infection of 10,000 people depending on initial battery levels

on any mobile platform with suitable exploits we present the implementation of the components needed for our concept malware for iOS. It makes use of the smartphone wireless features and exploits an iOS specific usability feature to be able to spread automatically. In lab experiments we measured the key values for our simulations. We showed by the comparison of existing models and our more complex one that more sophisticated simulation models have to be employed to evaluate the emerging threat of mobile malware. It is necessary to take movement and usage patterns as well as locations into account. One of the key findings in our simulation results is the fact that a critical mass of mobile devices will probably be reached in the near future. The monoculture of mobile operating systems combined with the large number of users will make mobile devices an attractive target for epidemically spreading malware attacks.

While the simulator already offers more relevant features to the spreading of mobile malware than most, there are still many improvements, which can be made to its realism. Further lab tests and user studies can be undertaken to fine tune parameters and thereby improving the predictive accuracy of the simulation. One major improvement will be a more detailed simulation of the interior of locations replacing the current indoor infection model. Also a wider range of user groups and device usage patterns is planed, allowing for devices to simulate Apps which each have their own usage patterns and can be targeted by further types of malware. Finally, the simulation of countermeasures for mobile malware attacks is another major goal that will be addressed in the future.



(a) 10,000 people, infections only on roads

(b) 10,000 people, all features

Fig. 9. Geo-spatial comparison of infections with and without locations

REFERENCES

[1] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," in *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, ser. SPSM '11, 2011, pp. 3–14.

[2] M. H. R. Khouzani, S. Sarkar, and E. Altman, "Maximum damage malware attack in mobile wireless networks," in *Proceedings of the 29th conference on Information communications*, ser. INFOCOM'10. IEEE Press, 2010, pp. 749–757.

[3] G. Yan, H. D. Flores, L. Cuellar, N. Hengartner, S. Eidenbenz, and V. Vu, "Bluetooth worm propagation," in *Proceedings of the 2nd ACM symposium on Information, computer and communications security - ASIACCS '07*, 2007, p. 32.

[4] P. De, Y. Liu, and S. K. Das, "An Epidemic Theoretic Framework for Evaluating Broadcast Protocols in Wireless Sensor Networks," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, Oct. 2007, pp. 1–9.

[5] S. Eidenbenz, "Modeling Propagation Dynamics of Bluetooth Worms (Extended Version)," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 353–368, Mar. 2009.

[6] K. Ramachandran and B. Sikdar, "Modeling Malware Propagation in Networks of Smart Cell Phones with Spatial Dynamics," in *26th IEEE International Conference on Computer Communications*, May 2007, pp. 2516–2520.

[7] H.-N. Nguyen and Y. Shinoda, "Modeling Malware Diffusion in Wireless Networks with Nodes' Heterogeneity and Mobility," in *Proceedings of 19th International Conference on Computer Communications and Networks*, Aug. 2010, pp. 1–8.

[8] C. Fleizach, M. Liljenstam, P. Johansson, G. M. Voelker, and A. Mehes, "Can you infect me now?" in *Proceedings of the 2007 ACM workshop on Recurring malcode*, ser. WORM '07, 2007, p. 61.

[9] K. Singh, S. Sangal, N. Jain, P. Traynor, and W. Lee, "Evaluating Bluetooth as a medium for botnet command and control," in *Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment*. Berlin: Springer-Verlag, 2010, pp. 61–80.

[10] Y. Bulygin, "Epidemics of Mobile Worms," in *2007 IEEE International Performance, Computing, and Communications Conference*, Apr. 2007, pp. 475–478.

[11] M. Martin, "An open source context simulator," Online: http://siafusimulator.sourceforge.net/, NEC Europe Ltd.

[12] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, S. Jajodia, and U. D. Milano, "On the Impact of User Movement Simulations in the Evaluation of LBS Privacy-Preserving Techniques," in *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, Malaga, Spain*, October 2008.

[13] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding the spreading patterns of mobile phone viruses." *Science (New York, N.Y.)*, vol. 324, no. 5930, pp. 1071–6, May 2009.

[14] J. Kim, V. Sridhara, and S. Bohacek, "Realistic mobility simulation of urban mesh networks," *Ad Hoc Netw.*, vol. 7, pp. 411–430, March 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1458725.1458838

[15] N. Husted and S. Myers, "Mobile location tracking in metro areas: malnets and others," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 85–96. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866318

[16] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta, "On cellular botnets: measuring the impact of malicious devices on a cellular network core," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 223–234. [Online]. Available: http://dl.acm.org/citation.cfm?id=1653690

[17] Symantec, "iphoneos.ikee," Online: http://www.symantec.com/security_response/writeup.jsp?docid=2009-111015-5423-99, November 2009.

[18] F-Secure, "Worm:iphoneos/ikee.b," Online: https://www.f-secure.com/v-descs/worm_iphoneos_ikee_b.shtml, 2009.

[19] D. Damopoulos, G. Kambourakis, M. Anagnostopoulos, S. Gritzalis, and J. H. Park, "User-privacy and modern smartphones: A siri(ous) dilemma," in *FTRA AIM 2012 International Conference on Advanced IT, Engineering and Management*, February 2012.

[20] P. James, "Intego security memo: Hacker tool copies personal info from iphones," Online: http://www.intego.com/mac-security-blog/intego-security-memo-hacker-tool-copies-personal-info-from-iphones, November 2009.

[21] D. Damopoulos, G. Kambourakis, and S. Gritzalis, "isam: An iphone stealth airborne malware," in *Future Challenges in Security and Privacy for Academia and Industry*, ser. IFIP Advances in Information and Communication Technology, J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, and C. Rieder, Eds. Springer Boston, 2011, vol. 354, pp. 17–28.

[22] K. Bauer, H. Gonzales, and D. McCoy, "Mitigating Evil Twin Attacks in 802.11," in *2008 IEEE International Performance, Computing and Communications Conference*, Dec. 2008, pp. 513–516.

[23] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proceedings of the first ACM conference on Wireless network security - WiSec '08*, Mar. 2008, p. 220.

[24] H. Gonzales, K. Bauer, J. Lindqvist, D. McCoy, and D. Sicker, "Practical defenses for evil twin attacks in 802.11," in *2010 IEEE Global Telecommunications Conference*, Dec. 2010, pp. 1 –6.

[25] B. Henne, C. Szongott, and M. Smith, "Towards a mobile security & privacy simulator," in *2011 IEEE Conference on Open Systems (ICOS2011)*, Langkawi, Malaysia, Sep. 2011.

[26] comScore, "comscore reports december 2011 u.s. mobile subscriber market share," Online: http://www.comscore.com/Press_Events/Press_Releases/2012/2/comScore_Reports_December_2011_U.S._Mobile_Subscriber_Market_Share, February 2012.

[27] A. K. Karlson, B. Meyers, A. Jacobs, P. Johns, and S. K. Kane, "Working overtime: Patterns of smartphone and pc usage in the day of an information worker," in *Pervasive*, 2009, pp. 398–405.