Design of a Hybrid Command and Control Mobile Botnet

Heloise Pieterse[1], Martin Olivier[2]

[1]*Defence, Peace, Safety and Security*
Council for Scientific and Industrial Research, Pretoria, South Africa
E-mail: hpieterse@csir.co.za

[2]*Department of Computer Science*
University of Pretoria, Pretoria, South Africa
E-mail: molivier@cs.up.ac.za

Abstract
The increasing popularity and improvement in capabilities offered by smartphones caught the attention of botnet developers. Now the threat of botnets is moving towards the mobile environment. This study presents the design of a hybrid command and control mobile botnet. The hybrid design explores the efficiency of multiple command and control channels against the following objectives: no single point of failure within the topology, low cost for command dissemination, limited network activities and low battery consumption. The objectives are measured with a prototype that is deployed on a small collection of Android-based smartphones. The results indicate that current mobile technology exhibits all the capabilities needed to create a mobile botnet.

Keywords: *Mobile, Botnet, Command and Control, Hybrid*

Introduction

The last few years saw a revolution in the development of cellular phones, transforming the devices from basic voice and text phones to all-in-one portable devices known as smartphones. Demonstrating functionality similar to that of a traditional computer, smartphone's today provide interconnectivity capabilities such as Internet access, device-to-device communication, and a wide variety of software applications. Improvement in smartphone capabilities and the popularity associated with mobile devices have caused malware developers to shift their focus towards mobile devices. During the first quarter of 2012 mobile malware increased by 1200% (Lardnios 2012). This sudden rise of malware coupled with the popularity of smartphones creates possibilities for new threats to emerge such as mobile botnets.

Botnets are a well-known threat to the users of the Internet and Personal Computers. They are responsible for the delivery of spam, collection of information, processing large quantities of data and causing distributed denial of service (DDoS) attacks (Grizzard et al. 2007). With the constant improvement of smartphone computing power and communication capabilities, malware developers are starting to introduce the concept of botnets to mobile devices such as smartphones. A mobile botnet is a network consisting of a collection of compromised smartphones, controlled by a botmaster through a command and control (C&C) network. The C&C network is the core of any botnet as it allows for the efficient dissemination of commands from the botmaster to all the bots. Traditional C&C technologies, such as those

based on HTTP, are also useful in mobile botnets. However certain smartphone capabilities, such as SMS and Bluetooth can provide the botmaster with additional C&C channels to support command dissemination. Given the popularity of smartphones and the continuous rise in mobile malware, it is only a matter of time before mobile botnets become a dominate force in the development of mobile malware.

This paper presents the design of a new mobile botnet, called the Hybrid Mobile Botnet, which exploits multiple C&C channels to disseminate the commands. The objective of this study is to explore the efficiency of multiple C&C channels and to illustrate that current mobile technology exhibit all the required capabilities for mobile botnet development and support. We also analyse the behaviour of the newly designed mobile botnet by building a prototype and deploying it on a small collection of Android-based smartphones.

The remainder of this paper is structured as follows. We discuss the history of mobile botnets, while the following section describes the model of the Hybrid Mobile Botnet. The design of the prototype, the execution and the results are presented. We determine whether the objectives of the Hybrid Mobile Botnet are achieved and the final section concludes the paper.

The History of Mobile Botnets

The history of mobile botnets does not date as far back as that of traditional botnets since mobile malware only started appearing in 2004. Still it took nearly five years before mobile malware displayed functionality that closely resembled that of botnets. The first was the Symbian worm Yxes (Apvrille 2010), which targeted Symbian phones running the OS9 operating system (OS). The malware was responsible for sending out SMS messages, retrieving the International Mobility Equipment Identity (IMEI) and the International Mobility Subscriber Identity (IMSI) numbers of the phone and communicating with remote servers. The ability of the malware to connect to the Internet was the key characteristic that had many believed it was part of a mobile botnet. The malware had no C&C network and although it had the ability to contact remote servers, the processing of commands were limited (Apvrille 2010).

Near the end of 2009 a new malware appeared that targeted Apple's iPhones. The malware, later named ikee.B (Porras et al. 2010), included C&C logic and allowed the botmaster to have complete control over the infected iPhone. To propagate, ikee.B, searched Internet IP addresses for SSH services and then attempted to connect to the responding service as root by using the default password, 'alphine'. The malware was responsible for archiving SMS messages and then forwarded the messages, along with other information collected from the phone, to a server located in Lithuania. Even though ikee.B had limited growth potential, it provided a foundation for the future development of mobile botnets (Porras et al. 2010).

During 2010 security analysts discovered a new Trojan horse, Geinimi, targeting smartphones running the Android OS. Geinimi (Wyatt 2012) is the first Android malware to display functionalities closely relating to that of botnets. The malware opened a backdoor on the infected device and transmitted the collected information to a remote location. It also has the potential to receive commands from a remote server. Besides the basic botnet functionality, Geinimi raised the sophistication of mobile botnet technology significantly. The malware deployed an off-the-shelf byte code obfuscator to hide botnet activities and encrypted chunks of the C&C traffic (Wyatt 2012).

The three versions of mobile malware described above established the platform for development of future mobile botnets. They revealed that it is possible to take concepts of botnets running on PCs and apply them to mobile devices. Indeed, it is possible to establish C&C of mobile botnets and the next section will discuss a hybrid approach.

Proposed Hybrid Mobile Botnet

The purpose of this new design is to explore the efficiency of using a hybrid structure consisting of multiple C&C channels to achieve a stealthy mobile botnet and also to illustrate that current mobile technology exhibit all the required capabilities needed to support a mobile botnet. In order to achieve a stealthy design, the Hybrid Mobile Botnet should consider the following objectives: no single point of failure within the topology, low (monetary) cost for command dissemination, limited network activities and low battery consumption per bot. The design of the Hybrid Mobile Botnet consists of the following three main components: propagation vector, C&C channels and mobile botnet topology.

Although multiple mobile botnet designs currently exists in literature (Geng et al. 2012; Singh et al. 2010; Xiang et al. 2011; Faghani & Nguyen 2012) our proposed mobile botnet is the first, to our knowledge, to use multiple C&C channels to disseminate commands. The use of multiple C&C channels makes this mobile botnet harder to detect, cost-effective and more reliable.

Propagation Vector

The propagation vector is responsible for disseminating the malicious bot code to the smartphones. Common techniques for spreading the code include social engineering or vulnerability exploits. The Hybrid Mobile Botnet exploits the method of social engineering by tricking users into downloading a popular application that is infected with the malicious bot code. Such an application is well-known and legitimate but the original code has been re-engineered and repackaged with additional bot code. A user installs the application but is unaware of the additional configurations taking place in the background of the smartphone.

For this propagation vector to succeed, the botmaster selects an application that is currently popular among smartphone users. In the following steps, the botmaster will reverse engineer the selected application and include malicious bot code without affecting the original code modules or their functionality. When the botmaster completes the repackaging of the application it is returned to the Application Market where it awaits downloading.

The motivation behind deploying this propagation vector is two-fold. Firstly, returning the malicious application to the Application Market provides this mobile botnet with the ability to reach a wide audience. Secondly, choosing a popular application also allows for the possibility that the malicious application can spread by word of mouth within social circles. It is because of these motivations that the Hybrid Mobile Botnet deploys via the propagation vector as described above.

Command and Control Channels

The C&C structure is the most important component of a mobile botnet as it is responsible for disseminating the commands from the botmaster to the mobile bots. Due to the critical aspect of the C&C channels, it forms an attractive target for a defender trying to bring the mobile botnet down. Therefore to achieve robustness and stealth the Hybrid Mobile Botnet follows a hybrid approach and will use a combination of popular C&C channels for mobile botnets,

SMS and Bluetooth, while also including HTTP. The purpose behind the selection of these three C&C channels are further explained in the upcoming sections.

SMS C&C Channel

SMS is a popular service offered by the mobile phone network and is supported by most smartphones available today. There are multiple advantages provided by SMS that makes it a suitable channel for C&C. These advantages include (Zeng, et al 2012):

- Ubiquity: Most smartphones can handle SMS messages.
- Offline accommodation: A Service Centre stores the SMS messages if the recipient's smartphone is turned off.
- Hiding malicious content: A SMS message can hide malicious content.
- Multiple send and receive channel options: For example sending SMS messages via online websites.

To design a stealthy unidirectional SMS C&C channel, the cost of sending the SMS messages and the prevention of the smartphone user detecting the received SMS messages must be taken into consideration. Currently there are services available that offer free SMS texting via web interfaces (for example Text4Free). Such websites offer the botmaster the opportunity to send multiple SMS messages without incurring any costs and possibly keeping his/her identity hidden.

To prevent the smartphone user from detecting the commands being sent as SMS messages, every mobile bot will intercept all incoming SMS messages before they reach the inbox. SMS messages containing the specific passcode will be aborted while all other SMS messages will safely pass through to the inbox to avoid any detection by the smartphone user.

Bluetooth C&C Channel

Bluetooth is the second C&C channel for the Hybrid Mobile Botnet. The reason for selecting unidirectional Bluetooth as a C&C channel is simply because of its availability on most smartphones and it also provides a stealthy mechanism for command dissemination. There is however an important aspect of Bluetooth that must be taken into consideration.

Bluetooth, like any other electronic component, consumes battery power. If the Bluetooth is left on indefinitely, it will quickly drain the battery of the smartphone which can lead to the discovery of the mobile bot. To minimize the consumption of battery power, the Bluetooth will only be active during specific period of the day and only for a limited time. These periods are known as periods of mobility and are defined according to Stability and Availability. For the purpose of this paper we defined three periods of mobility:

- No Mobility:
  Stability: Stability is high with no changes in geographical positioning.
  Availability: Active for long periods, during nightfall and early morning hours, when people are sleeping.
- Low Mobility:
  Stability: Stability is moderate with infrequent changes in geographical positioning.
  Availability: Active for moderate periods, during day time, when people are actively working.
- High Mobility:
  Stability: Stability is low with frequent changes in geographical positioning.
  Availability: Active for short periods, during morning hours and late afternoons as people travel to their destinations.

From the three periods of mobility mentioned above, the period of Low Mobility provides the most stable time period for the longest available time and therefore the Bluetooth C&C channel will only be active during this period.

HTTP C&C Channel

The botmaster requires knowledge about the mobile botnet and all of the mobile bots that are actively participating in the mobile botnet. To retrieve the required information, both the SMS and Bluetooth C&C channels are inadequate. Therefore the mobile botnet requires an additional channel to transport the information. The additional channel utilizes HTTP and it allows a mobile bot to transfer information to the Control Server.

The purpose of the bidirectional HTTP C&C channel is to forward information between a mobile bot and the Control Sever. The information include: mobile phone number, Bluetooth MAC address, geographical data, IMEI number and the IMSI number. Thus the HTTP C&C channel is purely there to support the construction of the ever changing mobile botnet.

Topology of the Hybrid Mobile Botnet

A mobile botnet consists of a collection of compromised smartphones that are organized into a structure, often referred to as the topology. The topology of the mobile botnet allows the botmaster to efficiently disseminate the commands to all the mobile bots currently participating in the mobile botnet. In order to describe the topology of the Hybrid Mobile Botnet, certain terminology must be clarified:

- Mobile bot: a compromised smartphone that can assume one of two distinct roles: cluster head bot or receiver bot.
- Cluster head bot: a mobile bot within a cluster botnet that directly receives the commands via SMS messages from the botmaster. It is also responsible for forwarding the received commands to the receiver bots in its assigned cluster.
- Receiver bot: a mobile bot that receives commands from a cluster head bot.
- Botmaster: the entity responsible for originating commands via SMS messages to a selection of cluster head bots.
- Control server: a server managed by the botmaster that stores information about the mobile bots actively participating in the mobile botnet.
- Active period: the time period that allows the cluster head bot to exchange commands with the receiver bots within its assigned cluster. Formed at a specific time and location.
- Bot ID: the Bluetooth MAC address of a smartphone.
- Bot list: each mobile bot contains a file that lists the Bot IDs of the other mobile bots within a specific cluster botnet.

The topology of the Hybrid Mobile Botnet is shown in Figure 1. For simplicity, the HTTP C&C channels to the Control Server are not shown.

The global botnet consists of a collection of cluster botnets and its structure is dynamic due to the constantly changing cluster botnets. The cluster botnet, which also forms a dynamic structure, consists of a collection of mobile bots in close proximity. The dynamic property of both the global botnet and the cluster botnets are due to the mobility of the infected smartphones.

To allow communication to occur within the cluster botnet, the mobile bots utilize the Bluetooth C&C channel. The Bluetooth C&C channel requires the mobile bots to be within close range (10 meters) to communicate and therefore the cluster botnet is also location dependent. Due to the location dependence and the dynamic property of cluster botnets, it will only exist for a specific time period at a specific location. Thus only during the available active periods will the cluster head bot exchange the commands via Bluetooth and will continue until all the receiver bots within the cluster botnet have received the command.

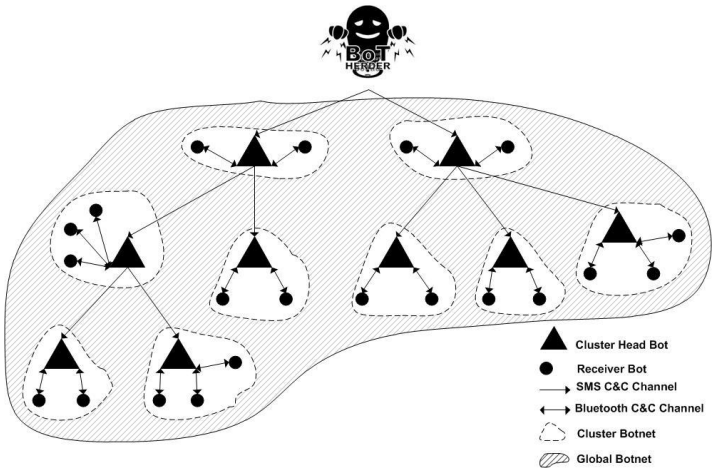


Figure 1: Topology of the Hybrid Mobile Botnet

With the Bluetooth C&C channel being location dependent, it is inadequate to use it as the global communication medium between the cluster botnets. Therefore the SMS C&C channel is used to propagate the commands via SMS messages to all of the cluster head bots. Due to the possible large number of cluster head bots it will be impractical for the botmaster to directly send the command to all of the cluster head bots. To keep monetary costs low, the SMS C&C channel utilizes an arbitrary tree structured topology. This arbitrary property of the tree structure provides the cluster head bot with the ability to send a specific number (between one and five) of SMS messages. The arbitrary tree structured topology improves the stealth of the mobile botnet and increases the difficulty of predicting the flow of command dissemination.

The dynamic topology increases the complexity of detecting the Hybrid Mobile Botnet, but it also complicates the process of command dissemination. Using the C&C channels as described above will allow the mobile bots to communicate in an effective and timely manner.

Prototype Design and Evaluation

The prototype consists of a small collection of smartphones, infected with malicious bot code and capable of running on the Android OS (version 2.3.3 and above). The purpose of this prototype is to evaluate the effectiveness of the Hybrid Mobile Botnet on real devices and also measure the objectives. The visualization of the prototype is shown in Figure 2.

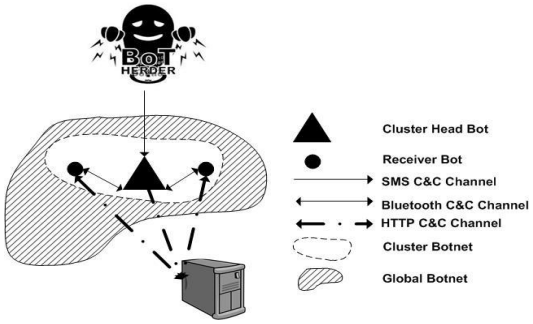


Figure 2: Topology of the Hybrid Mobile Botnet prototype

The prototype was specifically developed for the Android OS. The selection of the Android OS as the development platform is two-fold. Firstly, as of the second quarter of 2012 the Android OS is leading the market with 64.1% in smartphone sales, making it the most popular OS for smartphones (Van der Meulen & Pettey 2012). Secondly, besides the popularity, the Android OS also allows any user to create, develop and upload applications to Google's Play store. It is because of the above mentioned reasons that we selected the Android OS as the development platform.

The prototype consists of the following devices: Samsung Galaxy Pocket, Samsung Galaxy S2 and Google's Nexus 7 tablet. Each device is infected with exactly the same piece of malicious bot code. During the execution of the prototype, the battery consumption, data consumption and Anti-virus applications will be evaluated.

The prototype is designed to execute during hourly intervals (see Figure 3), instead of daily intervals, to easier evaluate performance and execution. With every hour interval the time period between twenty past and twenty to, for a total of 40 minutes, represents the period of Low Mobility and is the time when the mobile bots will be active. For the first two hours of execution, the prototype collects GPS data at ten minute intervals. During the last interval, the collected GPS data of that period of Low Mobility are uploaded to the Control Server via the HTTP C&C channel and the response from the Control Server will include the address of the elected cluster head bot and the command dissemination flag. The command dissemination flag indicates whether the following period of Low Mobility will perform command dissemination or will continue with the collection of GPS data. Only the botmaster can start the process of command dissemination by sending the command via a SMS message to the cluster head bot. If the response from the Control Server is positive for the start of command dissemination, then during the following active period the mobile bots will only focus on sending or receiving the commands. When the mobile bots complete the process of command dissemination and the execution of the received command, it returns to the collecting of GPS data.

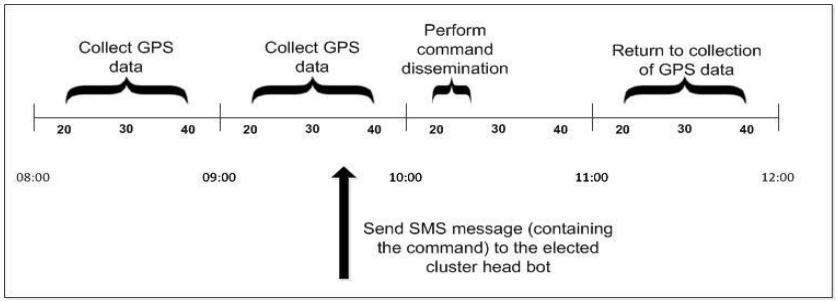
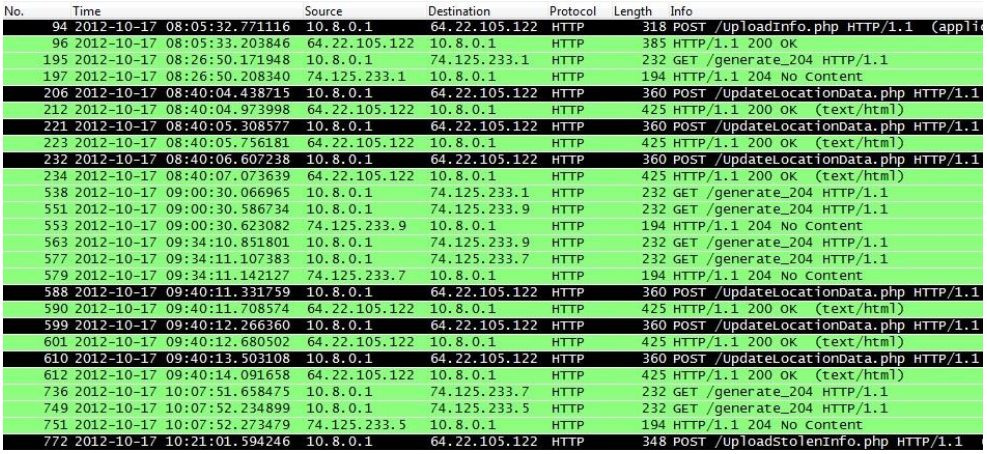


Figure 3: Timeline of execution of the prototype

Execution of the Prototype

To successfully track the execution of the prototype, the tPacketCapture application was installed on the Nexus 7 tablet (the only device to support the tPacketCapture application). This application performs packet capturing without the requirement of rooting the device. The captured data are saved as a PCAP file and can be viewed using Wireshark. Using this application makes it possible to capture the communication occurring between the mobile bot and Control Server. The Nexus 7 tablet will monitor the HTTP traffic that occurs across a Wi-Fi connection with this application.



No.	Time	Source	Destination	Protocol	Length	Info
94	2012-10-17 08:05:32.771116	10.8.0.1	64.22.105.122	HTTP	318	POST /UploadInfo.php HTTP/1.1 (appli
96	2012-10-17 08:05:33.203846	64.22.105.122	10.8.0.1	HTTP	385	HTTP/1.1 200 OK
195	2012-10-17 08:26:50.171948	10.8.0.1	74.125.233.1	HTTP	232	GET /generate_204 HTTP/1.1
197	2012-10-17 08:26:50.208340	74.125.233.1	10.8.0.1	HTTP	194	HTTP/1.1 204 No Content
206	2012-10-17 08:40:04.438715	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
212	2012-10-17 08:40:04.973998	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
221	2012-10-17 08:40:05.308577	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
223	2012-10-17 08:40:05.756181	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
232	2012-10-17 08:40:06.607238	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
234	2012-10-17 08:40:07.073639	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
538	2012-10-17 09:00:30.066965	10.8.0.1	74.125.233.1	HTTP	232	GET /generate_204 HTTP/1.1
551	2012-10-17 09:00:30.586734	10.8.0.1	74.125.233.9	HTTP	232	GET /generate_204 HTTP/1.1
553	2012-10-17 09:00:30.623082	74.125.233.9	10.8.0.1	HTTP	194	HTTP/1.1 204 No Content
563	2012-10-17 09:34:10.851801	10.8.0.1	74.125.233.9	HTTP	232	GET /generate_204 HTTP/1.1
577	2012-10-17 09:34:11.107383	10.8.0.1	74.125.233.7	HTTP	232	GET /generate_204 HTTP/1.1
579	2012-10-17 09:34:11.142127	74.125.233.7	10.8.0.1	HTTP	194	HTTP/1.1 204 No Content
588	2012-10-17 09:40:11.331759	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
590	2012-10-17 09:40:11.708574	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
599	2012-10-17 09:40:12.266360	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
601	2012-10-17 09:40:12.680502	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
610	2012-10-17 09:40:13.503108	10.8.0.1	64.22.105.122	HTTP	360	POST /UpdateLocationData.php HTTP/1.1
612	2012-10-17 09:40:14.091658	64.22.105.122	10.8.0.1	HTTP	425	HTTP/1.1 200 OK (text/html)
736	2012-10-17 10:07:51.658475	10.8.0.1	74.125.233.7	HTTP	232	GET /generate_204 HTTP/1.1
749	2012-10-17 10:07:52.234899	10.8.0.1	74.125.233.5	HTTP	232	GET /generate_204 HTTP/1.1
751	2012-10-17 10:07:52.273479	74.125.233.5	10.8.0.1	HTTP	194	HTTP/1.1 204 No Content
772	2012-10-17 10:21:01.594246	10.8.0.1	64.22.105.122	HTTP	348	POST /UploadStolenInfo.php HTTP/1.1

Figure 4: Captured data of the mobile bot's network activities

During the execution of the prototype, the Nexus 7 tablet acts as a receiver bot and only the receiver bots collect information from the device. Upon receiving the command, these mobile bots must collect the IMEI and the IMSI numbers of the device and send the information to the Control Server. For the purpose of this experiment, the IMEI and the IMSI numbers were simulated on the Nexus 7 tablet since the tablet had no mobile network connectivity.

The data captured while executing the prototype on the Nexus 7 tablet is displayed in Figure 4. The lines highlighted in black are the captured packets that directly relate to the execution of the prototype and will be looked at more closely in the remainder of this section.

The first connection made to the Control Server is via the UploadInfo.php script (see Figure 5). During this connection the receiver bot collects the mobile phone number and Bluetooth address of the smartphone and forwards it to the Control Server. The mobile botnet requires this information to uniquely identify each mobile bot.

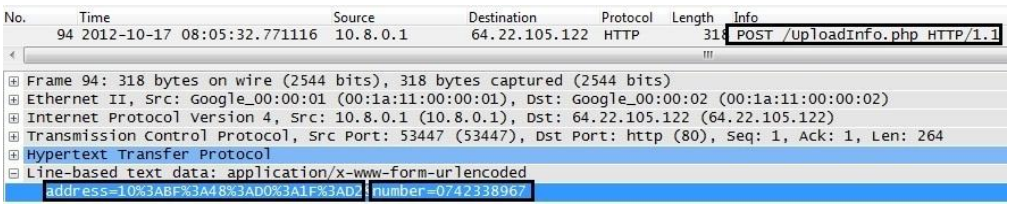


Figure 5: Captured data sent via UploadInfo.php script

Multiple connections occur between the receiver bot and the Control Server during 08:40:04 and 08:40:06 via the UpdateLocationData.php script. During each connection the mobile bot uploads the collected GPS data (see Figures 6, 7 and 8).

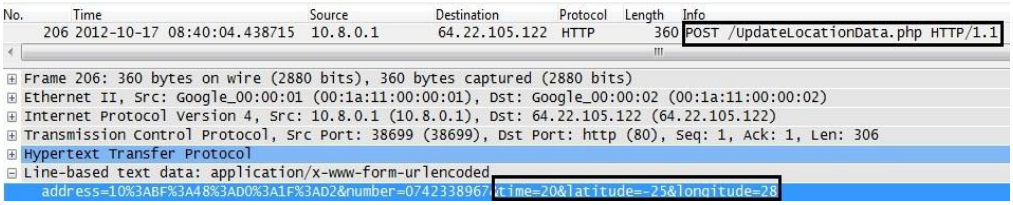


Figure 6: Collected GPS data for the 20 minute interval

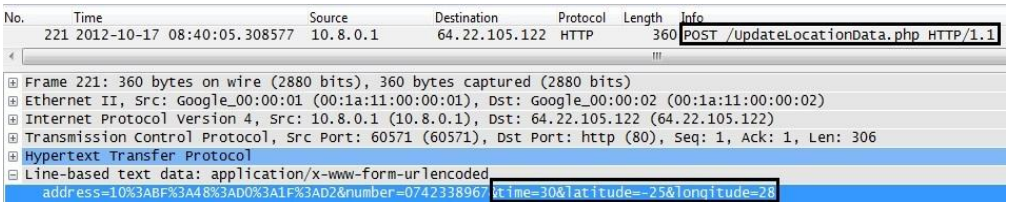


Figure 7: Collected GPS data for the 30 minute interval

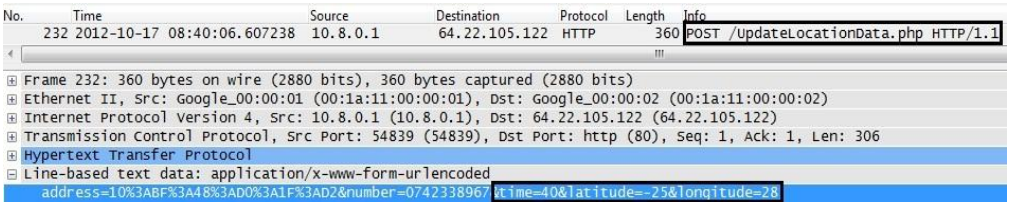


Figure 8: Collected GPS data for the 40 minute interval

After the last connection, the Control Server responds back with the Bluetooth address of the elected cluster head bot (28:98:7B:3A:8A) and the command dissemination flag (currently set to false). The false property of the flag means the following active period will continue with the collection of GPS data (see Figure 9).

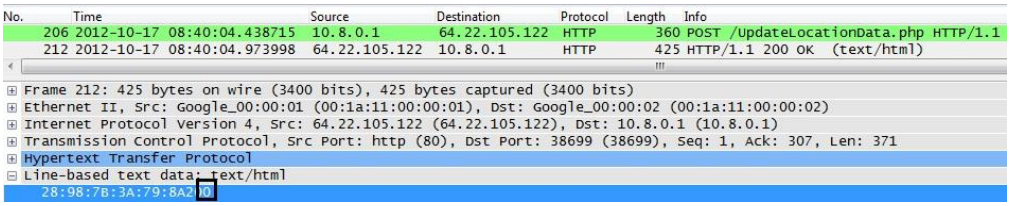


Figure 9: Response from Control Server

The next connections occur between 09:40:11 and 09:40:13, during which the collected GPS data is once again uploaded to the Control Server. The response received from the Control Server has however changed (see Figure 10). The command dissemination flag is set to true, meaning that the botmaster has sent the command via a SMS message to the cluster head bot somewhere between 08:40 and 09:40. Thus the next active period will perform command dissemination and execution.

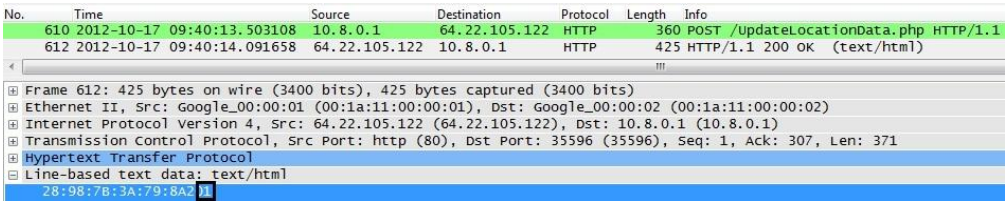


Figure 10: Response from the Control Server

Figure 11 reveals that the receiver bot has successfully received and executed the command. The mobile bot then forwards the IMEI and IMSI numbers to the Control Server via the UploadStolenInfo.php script.

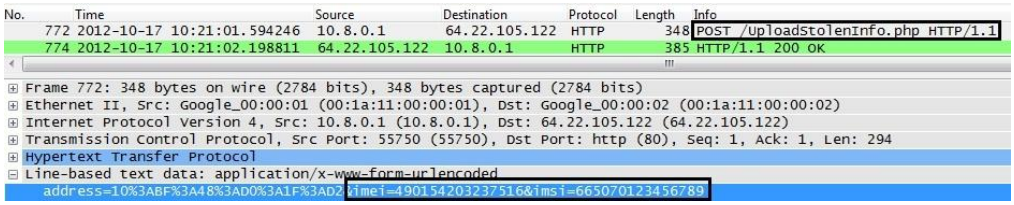


Figure 11: Stolen IMEI and IMSI numbers

This step-by-step analysis of the captured packets show that this prototype executed correctly, without any complications. This prototype thus illustrates that the current mobile technology exhibits all the capabilities required for developing a mobile botnet.

Evaluation

This section discusses the evaluation of the prototype on the mobile devices, focussing on battery and data consumption of a mobile bot, as well as the analysis of anti-virus applications. The prototype executes for a period of three hours and the same experiment was replicated on three separate occasions.

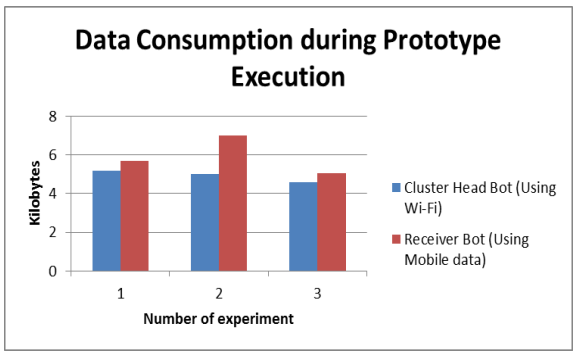
Battery Consumption

A significant decrease of battery power will potentially alert the smartphone user of the presence of the mobile bot which in return can lead to the bot's discovery. Therefore, throughout the execution of the prototype, the consumption of battery power was closely monitored.

To effectively monitor the consumption of battery power, the GSam Battery Monitor application was installed on all the smartphones participating in the execution of the prototype. This application can monitor each application or service individually and determine the consumption of battery power. During the three separate experiments, each mobile bot only consumed 0.1% of the battery power over a period of three hours. Thus during 24 hours the smartphone will only lose 0.8% of its power while the mobile bot executes. Thus the execution of the mobile bot will have little influence on the battery, increasing the difficulty of detecting the mobile bot.

Network Activities

A sudden increase in data consumption can potentially alert the smartphone user of the presence of the mobile bot. After the execution of the prototype, each smartphone was analysed to determine the data consumption of the mobile bot (see Figure 12).



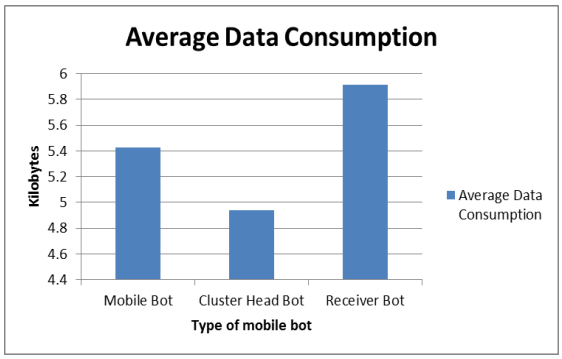


Figure 12: Data consumption of the prototype

On average, each mobile bot consumes 5.427KB during execution with a standard deviation of 0.846KB. Thus during a monthly period a mobile bot will consume less than 200KB of data. This low consumption of data will not alert the smartphone user and also increase the difficulty of detecting the mobile bot on a smartphone.

Anti-virus Analysis

Four mobile anti-virus applications (AVG Anti-virus, Avast Mobile Security, Lookout Security & Anti-virus and Norton Anti-virus & Security) were installed on a smartphone prior to the execution of the prototype. All the anti-viruses were active during the execution of the prototype but failed to identify any malicious activities. After the execution of the prototype, all of the anti-viruses performed a scan of all the available applications on the smartphone. None of these scans reported of any malicious application.

Also discovered during the evaluation of the anti-virus applications is the fact that they share most of the same permissions as the mobile bot application (Access location data, Read identity info and Access messages). So it becomes impractical to determine whether an application is malicious or not by simply looking at the permissions.

The analysis of four anti-viruses show that new malicious mobile malware can go undetected. This inability of the anti-virus applications to identify mobile bot activities and the sharing of multiple permissions improves the secrecy by which the mobile botnet can operate.

Discussion

The purpose of the prototype was to explore the efficiency of multiple C&C channels against the following objectives: no single point of failure within the topology, low cost for command dissemination, limited network activities and low battery consumption per bot. In terms of the first objective, the Hybrid Mobile Botnet accomplishes this by ensuring each mobile bot contains a file (Bot list) with all of the Bot IDs (Bluetooth addresses) of the other mobile bots within a certain cluster. Thus, if the Control Server should become unavailable the Hybrid Mobile Botnet will still be able to function to a certain degree by using the information in the Bot list.

By forming cluster botnets that can communicate to mobile bots in their assigned cluster via Bluetooth ensures that the overall cost of communication within the mobile botnet stays low. The limitation of the amount of SMS messages that can be sent from individual mobile bots also allows the cost of communication to be low, thus meeting the second objective.

From the evaluation of the prototype it is possible to conclude that the following objectives, namely limited network activities and low battery consumption, are met. Each mobile bot consumes on average 5.427 kilobytes while executing during the period of low mobility and by only connecting a limited number of times to the Control Server ensures that the objective of limited network activities are met. Only 0.1% of the smartphone's battery power was consumed during the execution of the prototype, confirming the compliance of the very last objective of this mobile botnet design.

This study demonstrates that a cost-effective and stealthy mobile botnet is possible through implementation of a hybrid architecture using existing smartphone communication channels.

Conclusion

As smartphones become more powerful, they become the ideal targets for mobile malware. One such threat that smartphone users are currently facing is that of mobile botnets. In this paper, we proposed the design of a new mobile botnet that utilizes multiple C&C channels, namely Bluetooth, SMS and HTTP. To analyse and measure the performance of this model, a prototype was designed and executed on a small collection of smartphones. From the analysis of this prototype, it is possible to conclude that this mobile botnet exhibits the following qualities: cost-effectiveness and stealth. Future research will focus on improving the mobile botnet design by encrypting all of the communication occurring between the mobile bots and the Control Server, randomizing the mobile bot activities so that it becomes even more difficult to detect and also exploring other possible C&C channels for command dissemination. The ultimate goal of this research is to discover techniques by which future mobile botnets can be detected on smartphones.

References

Apvrille, A. (2010), "Symbian worm Yxes: Towards mobile botnets?", *19th Annual EICAR Conference*.

Faghani, M.R. and Nguyen, U.T. (2012) SoCellBot: A new botnet design to infect smartphones via online social networking, *25th IEEE Canadian Conference on Electrical and Computer Engineering*.

Geng, G. Xu., G. Zhang, M., Guo, Y., Yang, G. and Wei, C. (2012) The Design of SMS Based Heterogeneous Mobile Botnet, *Journal of Computers*, 7(1): 235-243.

Grizzard, J.B., Sharma, V., Nunnery, C., Kang, B.B.H. and Dagon, D. (2007) Peer-to-peer botnets: Overview and case study, *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, USENIX Association.

Lardnios, F. (2012) McAfee: Mobile Malware Explodes, Increases 1.200% in Q1 2012, URL: http://techcrunch.com/2012/05/23/mcafee-mobile-malware-explodes-increases-1200-in-q12012/ [Accessed 8th October 2012].

Porras, P., Saïdi, H. and Yegneswaran, V. (2010) An Analysis of the iKee. B iPhone Botnet, *Security and Privacy in Mobile Information and Communication Systems*, 141-152.

Singh, K., Sangal, S., Jain, N., Traynor, P. and Lee, W. (2010) Evaluating bluetooth as a medium for botnet command and control, *Detection of Intrusions and Malware, and Vulnerability Assessmen*t, 61-80.

Van der Meulen, R. and Pettey, C. (2012), *Gartner Says Worldwide Sales of Mobile Phones Declined 2.3 Percent in Second Quarter of 2012*, URL: www.gartner.com/it/page.jsp?id=2120015 [Accessed 1st October 2012].

Wyatt, T. (2012) Security Alert: Geinimi, Sophisticated New Android Trojan Found in Wild, URL: http://blog.mylookout.com/blog/2010/12/29/geinimi_trojan/ [Accessed 4th July 2012].

Xiang, C., Binxing, F., Lihua, Y., Xiaoyi, L. and Tianning, Z. (2011) Andbot: towards advanced Mobile botnets, *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, USENIX Association.

Zeng, Y., Hu, X. and Shin, K.G. (2012) Design of SMS commanded-and-controlled and P2P Structured mobile botnet, *WISEC '12 Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, 137-148.