Antisocial Networks: Turning a Social Network into a Botnet

Lirim Osmani

Helsinki 20.04.2011
Seminar Paper
Security Testing

University of Helsinki
Department of Computer Science
Derived from the Czech word "robota", which literally means work or labour, initial concepts of bots did not include harmful behaviour by default. With the technology advancing, modern definitions of "bots" as zombies and drones were introduced. Attacks in 2007 against Estonia [Ott08], in 2008 against Georgia [Dan08], or in 2009 against South Korea [Kre09] have concluded once more that the industry is not prepared battling such malicious activities on a global scale. The magnitude of attacks performed and their malicious capabilities tells us that Botnets are continuously evolving in search of new communication vectors to propagate their malicious activities. In this paper we provide a comprehensive view of different Botnet technologies that have emerged in past few years. We present the basic architectures, discuss in brief few examples and we explore in more detail the new Botnet communication vectors emerging with the rapid development of social networks and their potential for malicious activities.

ACM Computing Classification System (CCS):
A.1 [Introductory and Survey],

Keywords: botnet, social networks,
Content

1 Introduction 1

2 Botnet Architectures 2
   2.1 Centralised C&C Architecture ................................................................. 3
   2.2 Decentralised C&C Architecture ............................................................... 4

3 The Dark Side of Social Networks 5
   3.1 Twitter ........................................................................................................ 6
   3.2 Facebook ................................................................................................... 7

4 Conclusion 10

References 11
1 Introduction

The birth of the "bot" can be considered 1989 when it was originally used for control instances located in the chat rooms of Internet Relay Chat (IRC). A bot initially was an automated script that had many good uses, such as "sit and listen" and take care of some generally harmless activity. At the time, the response to commands and security was not a concern. The primary characteristics of bots are that they autostart on reboot, they are always listening and always working, they can be downloaded and fully customised, and usually operate via remote IRC, TCP port or channel. Today the number of bots is on increase, some of them have quite a few advance install options like where they will install too, what they will do, what they will listen too, or how they will call out as well. The default service port for many bots is 6667 (relay-chat servers listen on ports 6000-7000), however with the technology progressing, bots today can use also p2p networks, thus becoming more and more dangerous because of their ability to infect and look like p2p traffic. Reports and analysis shows that Linux and Apple platforms are not typically targeted, and its more of a Windows problem than anything else. Usually the main characteristics of Botnets are infections on massive scale, this way making it so difficult to run test in live environments.

Capabilities of bots nowadays are such that they can be programmed to do about anything. They can perform DDoS attacks, secondary infect other systems, this way revealing their real danger as each bot infects its partner more and more bots get affected. The botcentric economy has matured in such a scale that nowadays we encounter Botnets for fun, for profit, some offer them for hire, and even worst people who have made a collection of bots over the years, have fleets of bots that you could have them do what you wish, spread illegal spam, perform ID theft and perform other phishing tactics. The primary targets are home users, universities, however nowadays companies and even countries are being targeted [Car10].

The Botnet life-cycle consist of preparing the bot (coding) and sending it to the wild. The bots get installed by the users by a virus, trojan, or exploit. Once connected to the intended IRC channel and the army has been created, the attacker (botmaster) can command and launch all sorts of malicious activities starting from attack this server, format HDD or whatever he wants. This gives the power to the attacker to control the
army of bots as he wishes and how he wishes.

First published in 1993 and further developed since, Eggdrop can be considered as the first IRC bot. Next, following the release of Eggdrop, malicious IRC bots appeared, created primarily in order to attack other IRC users or even entire servers. Shortly after, Denial of Service (DoS) and then Distributed Denial of Service (DDoS) were implemented in these bots. Tools like TrinOO, Stacheldraht and Tribal Flood Network 2000 made their way to scene with even more advanced options such as concentrated attacks from multiple sources [Eni11].

However with the client-server architectures dominating in the 1990s remote access tools like Back Orifice 2k or SubSeven were introduced as a prototype of Botnets with control over only one machine. The combination all of the aforementioned functionality with computer worms as their propagation vectors, finally resulted in the concept of early modern Botnets, with specimens like Pretty Park, GTBot, SDBot, Agobot, Spybot, Rbot and several more [Eni11]. In the next section we explore briefly the architecture models of the Botnets to proceed in more detail in section 3 with social networks and their implications for building Botnets for launching malicious attacks.

2 Botnet Architectures

In essence a bot is a passive entity until it connects to central server or any other infected machine know as the active or controlling entity. The bots provide a range of implemented features to a corresponding controlling entity. This entity is commonly know as the command-and-control (C&C) server usually under the control of one or more persons, called the botmasters who relay commands through this server. The C&C infrastructure typically serves as the only way to control bots within the botnet. The bots are required to maintain a stable connection within this infrastructure in order to operate efficiently. Taking this into account we make a classification of Botnet architectures into centralised and decentralised [Eni11].
2.1 Centralised C&C Architecture

In a centralised C&C infrastructure, all bots establish their communication channel with one, or a few single connection points, as illustrated in figure 1.

![Centralised C&C Architecture](image)

This enables the botmasters to communicate with the bots simultaneously and can issue commands to all the bots that are both online and connected to the botnet. Additionally, private conversations are possible on a one-to-one basis, thus allowing direct manipulation of single bots. Originating from the IRC protocol (which is text-based), it offers a robust and easy-to-implement approach into building and commanding a Botnet. IRC channels for Botnet control are either hosted on public IRC servers or on servers owned by the botmaster. If their own servers are used, arbitrary modifications to the protocol can be made using their own instruction sets and encryption. This strengthens the botnet against countermeasures for detection.[Eni11]

However with HTTP being currently the protocol most commonly used for the delivery of data over the Internet, has made HTTP available in nearly every network connected to the Internet and is rarely filtered. This goes in favor of Botnet operators, because it makes the protocol viable as a command-and-control protocol. Centralised command-and-control servers based on HTTP make up nearly 70% [Sym10] of all C&C servers and are therefore the most common way to control a Botnet. A typical example of Botnets using HTTP for communication are those generated with the commercial ZeuS crimeware toolkit [Bin10], allowing for the botnet to be managed even from a low technically skilled botmaster.
2.2. Decentralised C&C architecture

In the decentralised architecture bots employ the characteristics of p2p overlays. Namely, loosely coupled links between the bots enable communication within the Botnet and provide the basis for the organization of a new class of botnets known as peer-to-peer botnets.

Contrary to the centralised architecture, by introducing no concept of server and having all participants as peers connected to each other, the Botnet environment formed has all the participants exchange information with each other without the need of some centralized controlling entity. This attribute enables that even the information inside the Botnet is also shared evenly between the participants. Consequently, information about the whole Botnet cannot be obtained directly, and commands have to be injected into one peer of the botnet. The approach is usually directly over the communication protocol or even via the update functionality. The insertion of such updates and commands into the Botnet usually happens from an arbitrary point, making localisation of the botmaster almost impossible. With peer bots organising themselves into an overlay layered on top of the Internet protocol, provides a high degree of anonymity and the major advantage is that no central server can be attacked to mitigate them directly [Eni11].

![Decentralised C&C Architecture](image)

Fig. 2. Decentralised C&C Architecture [Eni11]
3 The Dark Side of Social Networks

Originating mainly from the IRC protocol we can say that botnets are currently in its evolution phase. Most security professionals and companies associate negative things with IRC and frequently block it. However with the explosion of social networking paradigm and its intrinsic property of having a large user database, it is quite common in a normal operational environment to expect a business machine to be active on Twitter, LinkedIn or Facebook. The open APIs on Twitter and Facebook provide a virtually unlimited resource for building target profiles. Twitter can already be used for tracking when at work, where we are going after work, or what we are doing now. Data mining, automatization and simulation of many social task inside the social networks has led to virtualizing communication to the degree that one cannot be certain of who he really is becoming friends with [Car10]. On the other side the statistics show that e-mail is not the top communication vector anymore and is falling behind social networking sites such as Facebook and Twitter. The operating systems and the software technology have become more mature, thus providing a higher level of security and stability to the users. With users pushing and pulling more and more content through the social sites everyday, has a created an ideal environment for the launch of different malicious activities. A new terminology as social zombies has emerged becoming as the next serious threat. With HTTP as the de-facto protocol for the delivery of data over the Internet, data uploads and downloads through the social networking sites are everyday activity in nearly every network connected to the Internet. They are rarely filtered as a traffic by system administrators, this way allowing the cybercriminals to explore new BotNet communication vectors for attacks. The industry again is not prepared and attacks have become a daily activity on a global scale. Popular sites such as Twitter, Facebook and LinkedIn are becoming increasingly attractive targets for spam, phishing, and malware.

In the next section we go briefly through the infrastructure that its commonly used in turning these sites into platforms for antisocial and illegal activities like DDoS attacks, malware propagation, spamming, privacy violation, etc. We present the main ideas, and mention some examples in the wild with more detailed focus on Facebook platform.
3.1 Twitter

With over 200 million users, Twitter is being labelled as the SMS of the Internet, becoming also the fastest growing social website for the moment. The concept is very simple: by allowing users to send read-text based posts composed of 140 characters called tweets, which are displayed on the users profile page, people interact with each other. Security flaws in Tweeter have been exploited many times already and they were covered in the media. The exploits were based mainly on early concepts that were not optimised and stealthy. However today the principal concept in exploiting the Tweeter platform is that the Botnet administrator can use Twitter, exclusively to manage the Botnet. The approach would be for the commander to send a tweet out as simple as bots do this and the bots would then listen to the command issued. The way it works is that one would be on Twitter and following someone that looks like a Botnet Administrator a Botmaster itself. The default syntax would be very basic: SYNTAX: colon, CMD, and then whatever we order the bot to do, for example: ping something, execute a command, or download a file. All done over Twitter. The idea has already been exploited so many times that there are even tools with the easy of use as “click and play”. Despi the fact that Twitter administrators are doing their best to block communication of that nature, the smart money would always respons with simply changing the language of communication. The state of play today is that languages a mostly incorporated as a snap-in module. You can write/create/drop in a module that does it in English. So instead of saying: SYNTAX :CMD PING 10.0.0.1 we could use instead look at this amazing address 10.0.0.1 and it will look like a normal twitter post. The rationale here is that the more efforts we put into the language the easier it would be to fit it to common tweet traffic. Intrusion Detection Systems will find this challenging to detect and Intrusion Prevention Systems will need modification to stop it [Hak10]. Althoug designed as a proof of concept, KreiosC2 [Kre] is such a tool dominating the Twitter platform. It has a modular design supporting many languages and channels. When it is up and running, the change of language and channel is a matter of command execution. The botmaster instructs the bots to change the language by simple saying ok bots, use this language instead, and then the bots go to a website, donwload the language file and changes to that language.
3.2 Facebook

Botnet command and control methods that utilize a JPEG file have been active for quite a while. Most people don’t know that JPEGs allow metadata. This attribute of the JPEG file format has enabled the adversarys to encode commands for actions into the metadata field. Once done, the bot then just needs to pull the same JPEG over and over again to keep things updated. As frequently or infrequently as needed, one could say, here are 20 JPEGs, go grab all of them. This activity is perceived as normal on Internet and is not going to trip any IDS, or any other perimeter defense for that matter[Ha]^10. This simple concept has led to the development of many malicious activities even at present. Given the number of users currently facebook has and the growing rate, by no doubt that it has been a targeting platform for numerous malicious activities.

In the next paragraph we review a concept already developed and tested by [Ath08] that proved to be succesful in utilizing the facebook platform for malicious activities. As authors indiciate in their paper, the experimental setup was to developed a proof of concept FaceBot for demonstration purposes. A real world application called the "photo of the day" was created that presented a different photo from National Geographic to facebook users everyday. Everytime a user clicks on the photo of the day application and image from the service of National Geographic appears [Nat]. However the application had a special code embedded that the user was not aware that is activated everytime a user reviews a photo. Namely the application code embeds four hidden frames with inline images hosted at the victim.

Despight the fact that facebook authorites highlighted that such experiment would require a lot of resources [Ley08], the team still conducted the experiment with their own lab setup and presented the results associated with the traffic experienced by their server that was configured as a victim host. By analysing attack magnitude, distribution and firepower the authors presented findings that could be used as measurement template for future malicious activities. The approach was:

**Attack magnitude:** By conduction measurement over the period of two weeks (fig2) and during the fixed hours (17-21) of Internet usage, it was noticed that the traffic pattern is quite bursty.
This underlines the social nature of the attack platform where users come and go in bursty fashion (approx. at the same time). The conclusion was that a malicious Facebook application can absorb Facebook users and force them to generate HTTP request to a victim host in burst mode fashion. By taking into account the timings and the periods an adversary could employ a more sophisticated technique and create a JavaScript snippet that continuously requested documents from a victim's host over the given time. Such attack could prove to be very malicious if planned carefully with regards to a particular event happening on a particular day.

**Attack distribution:** by using the GeoIP tool[Geo] and the IP addresses from the logs of the victim server that was setup, the origin of requests proved to be on a global scale.

As seen in fig. 4 from visualizing the data, they market with black every country from which they have recorded at least one request. It is evident that the nature of a FaceBot, even one that is a proof of concept, was highly distributed globally.

**Attack Firepower:** the firepower of a botnet proved to be an important concept for analysis. Practically it determines how powerful a botnet can be in performing its
malicious activity. A mathematical model was constructed by using simple parameters
such as the distribution of outgoing traffic, outgoing traffic of the facebook application,
and the number of users accessing the application over a time period. Assuming that
the adversary has developed a higly popular facebook application, which employs tricks
presented in the previous section the approach was:

We denote with $F(t)$ the firepower of botnet which is essentially the distribution of
outgoing traffic towards the victims server. Hence the aim is to find an analytical
expression for $F(t)$. Let $a_{out}$ be the outgoing traffic a facebook application can
pull from the victims host once the user is tricked into using the malicious application,
and further with $U(t)$ we denote the number of users accessing the application over
time. Thus we can formulate the following:

$$F(t) = a_{out} U(t)$$

(1)

$U(t)$ is function over time and given the nature of the value it represents, it will
have to incorporate the number of active users over the period $P$ $u(t)$ and an estima-
tion of user’s inter-arrival times $u_r(t)$, then:

$$U(t) = \frac{\int_0^P u(t) dt}{u_r(t)}$$

With the parameters identified, and under the assumption that there is a Facebot build
based on a popular facebook application, the firepower of such Facebot at the time $T$
results in:

$$F_T = a_{out} \frac{\int_0^P u(t) dt}{u_r(t)}$$

for example, if we have a FaceBot with $a_{out} = 10 \text{Kbit/sec}$ which is installed by 1,000
users, from whom 100 were active in the period of 10 seconds and their average in-
Theoretical examples presented in [Ath08] indicated that the firepower formula derived can be easily used in a more general sense. By looking at the adonimics.com [Ado] top application list one can easily calculate the firepower of a potential botnet if any of the applications on the list was used as a Facebot. To an adversary this approach will ensure more analytical and better planning for forming the Botnet.

4 Conclusion

Having a “one stop shop” solution for mitigating attacks generated by Botnets still remains a challenge. As seen already, Botnets are becoming more complex with different properties and attributes making them more adverse in their malicious activities. More advanced Botnets appear to be a collection of malicious tools grouped under single umbrella that make the analysis for the researchers even harder. As suggested by [Eni11] a integrated approach of mitigating the existing and preventing new infections should be considered in battling such activities in today’s modern Internet. However still, as Web 2.0 services are growing in numbers and gaining a certain level of acceptance among enterprises, cybercrime innovation is progressing in fast pace. Most social networking sites such as Twitter, MySpace, LinkedIn and Facebook, have been exploited already and used as surrogate C&C infrastructure. By utilizing the technology already available these social forums have been configured to issue obfuscated commands to globally distributed botnets. As described in [Ath08] the main attraction to these sites and services lies in the fact that they offer public, open, scalable, highly available, and relatively anonymous means of maintaining a C&C infrastructure, which further reduces chances of detection by traditional antivirus technologies. As we showed experimentally that the victim of a FaceBot may be subject to an attack that will cause it to serve data to the magnitude of gigabytes per day.

Sites like Tweeter and Facebook with hundred of millions of users already, are hosting applications that live inside their network. Attracting a large user-base as described can be done easily, and very quickly can be redirected to attack a victim host. As the results show the distribution of bots was on an global scale making it harder for detec-
tion and prevention. Providers of social platforms should be careful designing their platform APIs and offer them to the public for future development. Experience shows that to improve security and stability, the interactions between the social utilities they operate and the rest of the Internet should be minimized[Ath08]. This would encounter:

- strict API for giving access only to resources related to the system;
- every application should run in isolated environment;
- prevent the applications from interacting with other Internet host which are not participants of the social network;

As the technology progresses so will the innovations in Botnets and launching malicious activities. Botnets are already taking advantage of p2p communication, update, and management channels. With the cloud services as the next promising technology its expected that the C&C functionality will be more effectively dissipated. Spamming capabilities will further be enhanced. Botnets such as Koobface, KreiosC2 already use social networking services for propagation by sending out messages and by writing malicious posts on user's walls. We can thus fully conclude that *what viruses were for the last 20 years, Botnets will be for the next 20 years* [Hak10].

References


Dan08 Danchev, D., Coordinated Russia vs Georgia cyberattack in progress. ZD-


