2014 6th International Conference on Cyber Conflict
P.Brangetto, M.Maybaum, J.Stinissen (Eds.)
2014 © NATO CCD COE Publications, Tallinn

Botnet over Tor: The Illusion of Hiding

Matteo Casenove
VrijeUniversiteit
Amsterdam, The Netherlands
m.casenove@gmail.com

Armando Miraglia
VrijeUniversiteit
Amsterdam, The Netherlands
a.miraglia@student.vu.nl

Abstract: Botmasters have lately focused their attention to the Tor network to provide the botnet command-and-control (C&C) servers with anonymity. The C&C constitutes the crucial part of the botnet infrastructure, and hence needs to be protected. Even though Tor provides such an anonymity service, it also exposes the botnet activity due to recognizable patterns. On the one hand, the bot using Tor is detectable due to the characteristic network traffic, and the ports used. Moreover, the malware needs to download the Tor client at infection time. The act of downloading the software is itself peculiar and detectable. On the other hand, centralized C&C servers attract a lot of communication from all the bots. This behaviour exposes the botnet and the anomaly can be easily identified in the network.
This paper analyses how the Tor network is currently used by botmasters to guarantee C&C anonymity. Furthermore, we address the problems that still afflict Tor-based botnets. Finally, we show that the use of Tor does not, in fact, fully guarantee the anonymity features required by botnets that are still detectable and susceptible to attacks.

Keywords: *Botnet, Tor, Command-and-Control, Malware, Anonymity, Resilience.*

1. INTRODUCTION

Nowadays, one of the main threats that the Internet users face are *botnets*. Botnets are employed for many kind of malicious activities; examples are DDoS, personal data theft, spam, bitcoin mining, and cyber-espionage [19][9]. In the last ten years, the main antivirus vendors have reported a constant growth of botnets in the wild [1][2].

Traditionally, botnets are centralised overlay networks where the Command-and-Control (C&C) servers act as single point of control. Centralised botnets are easy to manage and maintain due to their centralised structure. A botmaster has a clear overview of the overlay network and she manages the bots, which, in turn, connect to the C&C servers to be reachable. Nevertheless, this architecture has an important drawback: the C&C servers are exposed and represent a single point of failure. Hence, by taking down the C&C servers, the whole botnet is defeated. In order to overcome this problem, botmasters have moved to more resilient unstructured P2P

networks for their bots. In this manner, P2P botnets remove the single point of failure, building a completely distributed network. In this network structure, the bots exchange commands among themselves. Ultimately, this new architecture achieves resiliency, making the disruption of the botnet significantly harder.

Alternatively, some botmasters have opted to keep the centralized structure of the botnet but, at the same time using improved techniques to decrease C&C servers detectability. In fact, the simplicity of the protocols and of the network organization are desirable properties. One of the most interesting techniques to achieve this goal is the use of the *Tor network*. By means of the Tor network, the botmaster can anonymously locate their C&C servers, which, in turn, are contacted by the bots which join the botnet. Tor is a network that provides anonymity. It creates an encrypted routing system to avoid traffic analysis and allows to publish services without revealing their locations. To do so, Tor provides the so-called *hidden services*. Hidden services [21] are characterised by services like web servers, shell providing services and others which are accessible only via Tor. In this manner, the client using the service does not require the actual address, and hence the actual location of the service, guaranteeing the service anonymity. In turn, botmasters can configure the C&C servers as hidden services. In this way, it is not possible to detect the C&C locations and, consequently, take down the botnet. Additionally, even though this technique is yet not being actively deployed for P2P botnets, it is still fully applicable and could, in fact, provide further resilience to take down attempts. Unfortunately, while the botmaster tries to hide her network, she also exposes it due to peculiar properties of the Tor network.

In this paper, we describe the weaknesses, overlooked by botmasters, which derive from the deployment of botnets over Tor when aiming for stealthiness. We present the use of Tor by the botnets providing the following contributions. Firstly, (a) we argue that botnets over Tor are still subject to the very same attacks used to defeat botnets that do not use Tor. Afterwards, (b) we argue that in some situations moving to Tor is counterproductive for the botmaster since this creates an anomaly in the normal network flow, attracting the observers attention. Finally, (c) we discuss the vulnerabilities of the Tor network that can expose the botnet, and mine the anonymity offered.

The remainder of the paper is organised as follow. In Section 2, we provide an overview of the background knowledge required. Section 3 describes the use of Tor in real-life botnets. After presenting the current state of these botnets, in Section 4 we analyse the vulnerabilities of this approach. Moreover, in Section 5 we discuss the related work and finally Section 6 summarises our work and provides a conclusion of the paper.

2. BACKGROUND

In order to better understand the problems that arise from building botnets over Tor, we first clearly describe the way in which botnets are structured and what type of features they use to achieve resiliency. Secondly, we present the Tor infrastructure and all its actors involved in the management of the system. These are crucial concepts, which are required before analysing that the combination of these two strong systems does not necessarily produce a stronger one.

A. Botnets

A Botnet is an overlay network of compromised machines called bots that are controlled by an attacker (botmaster). In order to connect the bots together, the machines are infected with a malware. There are different ways to infect a machine such as *0-day* exploits and spam. However, the most effective method used nowadays is *drive-by-download*. Using this last vector of infection, the attacker compromises a site which, in turn, is visited by a user. When visiting the page, the user will unknowingly download and install the malware, hence becoming infected. The botmaster uses the botnet to control the bots. By issuing commands, she can instruct the bots to perform malicious activities, namely *DDoS*, spam campaigns, credential theft, cyber-espionage, bitcoin mining, and others.

Botnets can be distinguished based on (a) the kind of malicious activities they perform, (b) the protocol they use, and (c) their architecture. Traditionally, the botnets have a very basic structure where every bot is connected to a central server controlled by the botmaster. In order to simplify the control of the bots, botmasters have deployed botnets with IRC-based (Internet Relay Chat) communication. The central server is also called C&C. This structure makes the botnet very easy to control but also easy to attack. The C&C is the only central node that connects all the bots. Hence, by taking down the server, the whole botnet is disrupted. For the last years, botmasters have continually updated the protocol and the architecture of their botnets. Botmasters have replaced a single C&C with multiple C&C servers or have used the *Fast-Flux* technique in order to achieve a better resiliency. They have also implemented *Domain Generation Algorithms* (DGA), which allow the malware to generate a random domain name at runtime. The random generated domain locates the C&C server[1]. Despite all the efforts from the botmasters, the centralised structure makes the botnets weak and easy to take over. In fact, the central C&C remains the single point of failure.

Since the structure is the weakness of such botnets, botmasters have moved to a more resilient structure: the P2P network [18]. The P2P architecture replaces the central C&C with a completely distributed network of bots. Bots exchange information between each other, transmitting commands and overlay management information using custom protocols. They also use common protocols, such as HTTP, DNS, and others, in order to be as stealthy as possible for operations like downloading new versions of the malware. Of course, this also makes the botnet more difficult to manage and to monitor. The P2P structure makes the botnets more resilient but not invulnerable against attacks or even disruption [19]. The P2P botnet can be identified using a method called *crawling*. With crawling it is possible to enumerate all or almost all the bots in the botnet. Furthermore, disruption can be achieved using *sinkholing*. Sinkholing is a technique that allows disruption by injecting crafted information in the list of peers of every bot. This modifies the structure of the network, turning it into a centralised network. The injected node can be controlled by the defender or can be inexistent, making all the bots point to a black hole.

Crawling and sinkholing are not always successful, since they require a deep analysis of the botnet protocol. Additionally, botmasters improve their botnets over time to prevent these attacks. Currently there is a competition going on between botmasters and researchers where botmasters try to make the botnets resilient and powerful while the researchers try to take them

1 Contextually, the botmasterhas to register the samerandom domain name and link with the C&C server.

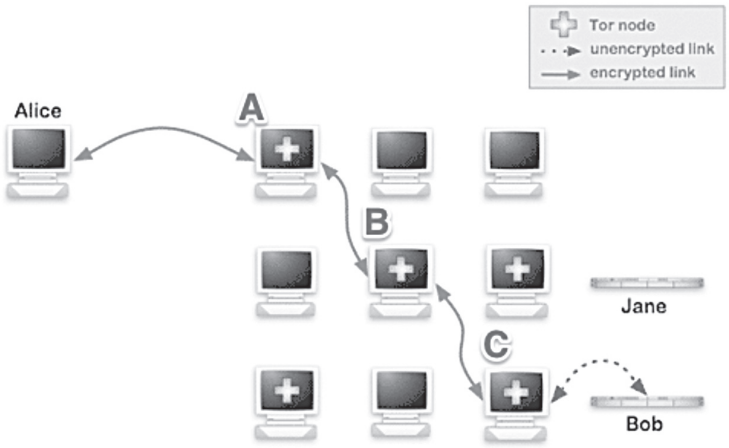
down. Lately, a new trend is arising: botmasters have started using Tor to hide the C&C servers [4] [5] [8].

B. Tor: Third-generation Oniuon Router

In a world where governments make use of monitoring and censorship, Tor [20] allows users to evade these invasive governments activities by providing anonymity. It is a network of volunteers, which provide thousands of relays used to route communication in the Tor network. The onion name comes from the multilayer encryption used by the relays in order to provide confidentiality. In fact, the encryption protocol is designed to avoid giving the relays access to the data they are routing.

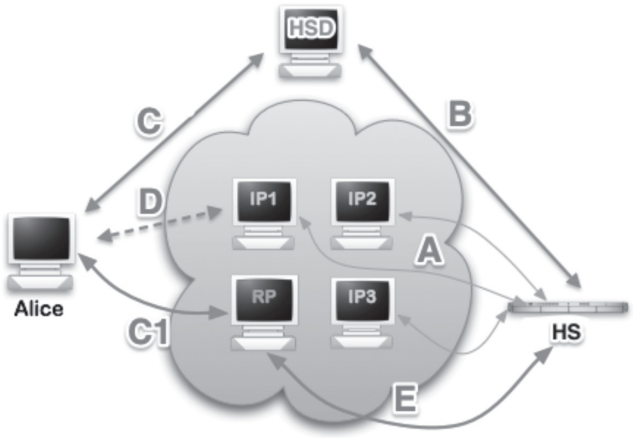
Consider the scenario where Alice wants to communicate with Bob in an anonymous way as in Fig. 1. At first, Alice randomly selects three different relays, which represent the entry point (A in Figure 1), the exit point (C in Figure 1) and the middle hop (B in Figure 1), creating the so-called *virtual circuit* from the source to the destination. The client negotiates with each relay in the circuit a separate set of encryption keys in order to enhance the privacy on the circuit. In such a scenario the flow of the communication evolves as follows. Alice sends the message to the entry point using an encrypted channel, then once inside the Tor network, the message is sent from relay to relay until it reaches the exit point where the communication is sent in clear to the destination.

FIGURE 1: TOR PROTOCOL.



Because each relay sees no more than one hop in the circuit, there is no way for a malicious relay or a monitoring system to trace the communication from the source to the destination. Since it is the exit point that sends the message to the destination, Bob does not know who the real source is but he only sees the exit point.

FIGURE 2: TOR HIDDEN SERVICE PROTOCOL.



Tor also allows to publish services inside the network anonymously (Figure 2). These services are called hidden services (HSs). Mainly, a service needs to publish its presence in the network in order to be reachable. The HS selects a set of relays asking them to be its introduction points (A in Figure 2). It creates a hidden service descriptor containing its public key and the address of its introduction points, then it inserts the descriptor in a DHT using an address like *XYZ.onion* as key (B in Figure 2). The *.onion* address can be used by the client to contact the HS (C in Figure 2). The DHT is implemented in Tor using the Hidden Service Directories (HSDs). With the *.onion* address, the client downloads the descriptor and it creates a new virtual circuit to a random relay asking it to act as rendezvous point (C1 in Figure 2). The client then uses the introduction points to inform the hidden service about the rendezvous point (D in Figure 2). Finally the HS creates a virtual circuit to the rendezvous point and starts the communication with the client (E in Figure 2).

This standard design of Tor is vulnerable to the traffic confirmation attack [3] that permits an attacker to confirm that two parties are communicating by observing patterns. If the attacker controls the edges of a circuit, it can expose the hidden service. In order to avoid such a scenario, Tor introduces the concept of *guard nodes* [21]. The entry guards are special relays, flagged as "good enough" to be guard, selected at random to become the new entry point in the circuit.

Unfortunately, even using the entry guards, Tor still has some vulnerabilities [17] [11]. In fact, it is not feasible to build such a big and complex system, which is completely secure. Therefore, Tor is not claimed to be foolproof or to provide absolute anonymity but it is instead guaranteed to provide an anonymity service, which is "good enough".

3. BOTNET OVER TOR

Since 2010, the use of Tor to hide botnets infrastructures has been discussed. In particular, the famous presentation by Dannis Brown at *DefCon18* [4] has shown, for the fist time, a possible implementation of a C&C channel over Tor to provide C&C server anonymity. Even after the presentation, we have not seen real application of this idea until Guarnieri in [5] detected and analysed the first Tor-based botnet. Announced on Reddit, the botmaster published the following message: "Everything operating tru TOR hidden service so no feds will take my servers down.". The botnet is a modified version of Zeus with DDoS, bitcoin mining and credential theft capabilities. The malware contains the Zeus bot, the tor client, the GMinerbitcoin mining tool, and few libraries for GPU hash cracking. All the bots run inside hidden services and all the C&C communication happens inside the Tor network. Avoiding the use of the exit nodes, the botmaster tries to reduce the botnet traceability. It uses an old-style IRC protocol to communicate with the bots and to issue commands. To be ethically correct with the Tor philosophy, the botmaster also makes the bots act like relays enhancing the Tor network while exploiting it.

Late summer 2013, a post on the Tor mailing list [6] raised the attention on a huge increment of the network usage and the amount of users in a really short amount of time. At the beginning no one could explain such an atypical situation [7] but then researchers [8] discovered that it was caused by a very big botnet that suddenly switched to Tor. The botnet uses the HTTP protocol over Tor with a centralised structure. It uses a preconfigured old version of Tor to connect to the network.

Unfortunately, this caused problems to the Tor network due to the significant increase of Tor communication going through the relays. In fact, the computational overhead caused by the expensive encryption operations has reduced the responsiveness of the system. This made many people unhappy and raised a lot of discussion especially by the Tor users [7].

These two examples show that botnets over Tor are no longer a forum discussion but have become a reality. The main reason that motivates botmasters to move to Tor is to find a new environment to achieve stealthiness and untraceability. The Tor hidden services provide anonymous C&C servers, which are more difficult to take down. Even attacking the server with a DDoS attack is unfeasible because the whole Tor network would be under attack. However, overconfidence can be dangerous.

4. THE FAILURE OF STEALTHINESS

Seeking for resiliency, botmasters are continuously evolving their botnets in order to resist against attacks to their network. Nowadays, P2P botnets show an improved resilience [9] but still Tor represents an appealing environment for botnets. In fact, to reach such resiliency the P2P botnets apply quite sophisticated techniques, while Tor provides anonymity and resiliency also for centralised infrastructures, which require less effort.

Unfortunately, this is not completely true. We argue that the botnets over Tor are interesting solutions but not as perfect as expected. Tor-based botnets do not represent the ultimate stage of resilience and are not less affected by the same vulnerabilities. For example, P2P botnets over Tor are not yet present in real life but within the bounds of possibility and it would be interesting to analyse the impact of Tor on their resilience. Every bot runs inside Tor as an HS creating an overlay network on top of the Tor network. The bots are identified by .onion addresses and they communicate using the classic custom protocols but this time tunnelled in Tor.

Surely, these botnets are not less subject to the very same kind of attacks applied to standard botnets. Even though bots are running as HSs and so their identities cannot be revealed, the crawling attack is still applicable. It would require only to use *.onion* addresses instead of normal IPs. Crawling aims to enumerate the bots in the network and, using Tor, we can enumerate the *.onion* addresses, which are part of the botnet. In this case, the use of Tor addresses gives the crawling an important advantage. While in standard networks we risk to overestimate the size of the botnet due to dynamically assigned IPS, the *.onion* addresses are uniquely assigned to each hidden service. The addresses are linked to the keys of the hidden service and do not change over time[2]. Hence, crawling becomes much more accurate when using such addresses. As a result, the technique can almost exact estimate the botnet network size.

The sinkholing attack injects fake nodes in the peerlist of the Tor-based bots as well as it does for the normal bots. While in the standard network, a sinkhole would be a standard IP address, in the Tor network the injected address would be a hidden service *.onion* address. Tor itself does not add any extra security feature against this attack, since it is a result of flaws of the botnet protocol and not of the network used. Furthermore, in Tor we face the ethical limitation based on which we cannot inform, attack or disconnect the bots in a P2P botnet. Geographically locating or identifying the IP of the bots is not really useful in a P2P botnet, since we cannot apply any direct action on the bot itself. For this same reason, crawling inside the Tor network has the same final effect of the normal crawling.

In the centralised botnets, the use of Tor can give an immediate solution for the single point of failure (e.g. if we cannot locate the C&C server we cannot take it down). Even attacking the server can be tricky. For example, by DDoS-ing the server we attack the whole Tor network [10]. Unfortunately, Tor has vulnerabilities that can result in the deanonymisation[3] of a service. Biryukov et al. [11] describe the possibility to determine the IP address of a hidden server exploiting the use of the guard nodes. Moreover, the Tor network is vulnerable to the traffic correlation attack where an attacker controls one or significantly many relays [13]. This is not an unrealistic scenario especially considering the recent data gate scandal [14] where the NSA was monitoring almost every communication channel in countries like USA. When we have a P2P botnet over Tor, it is very difficult to deanonymise every bot in the network but when we have a single or even few C&C servers it becomes feasible. This means that even using Tor, a centralised botnet has the same source of vulnerability, namely the single point of failure.

2 The .onion address does not change over the time unless the HS explicitly reboots itself and intentionally recreates the keys. In this way, a HS appears as a new service and it has to register itself inside the Tor network. This results in a really expensive operation and highly unlikely to be performed by a bot.

3 Deanonymise a service in the Tor network means that the real IP address of the service is revealed and so Tor does not provide anonymity anymore.

Lastly, when a botnet takes part in the Tor network, it raises a lot of attention because it creates instability in an almost stable network as Tor is characterised by a slow variation in the number of nodes. A botnet is often comprised by million of bots and when so many nodes join the Tor network in a short time, it signals that something wrong is going on. Mimoso [15] points clearly that a botnet undetected for years suddenly decided to hide the C&C on Tor and at the same time exposed itself making its presence more obvious.

A botnet using Tor leaves traces even from a client point-of-view. The way Tor is used nowadays by the malware has nothing to do with stealthiness. In fact, malware currently runs the Tor client as an external process. If the client was not previously installed, the exposure of the malware activity would be trivial. In fact, by verifying the list of running processes, the malware would be detected by identifying the Tor client process. Even though this is a clear symptom of infection, bot writers have not deployed any hiding technique.

At this point, it is clear that Tor does not provide the botnets with the expected capabilities, at least in the way it is currently used.

5. RELATED WORK

While research has studied botnet identification and analysis, no focus has been put into analysing botnet over Tor.

Rossow et al. [16] describe the different techniques that the botmasters apply to create resilient P2P botnets. They present an analysis of the resiliency of different families of P2P botnets against classic attacking methods such as crawling and sinkholing. They show which level of disruption can be achieved using these kinds of attacks and which families are more subject to them. This work gives us an understanding about the increment of resilience produced by the P2P structure but also suggests how this new structure is still attackable.

Andriesse et al. [9] make an in depth analysis of the latest state of the art in resilience in P2P botnets, in particular for the Zeus botnet. In this paper, they dissect the last version of the Zeus protocol describing the algorithm used and the resilient features applied. This work shows how strong and resilient the latest P2P botnets already are without using Tor.

A lot of research has been also done with respect to the security in Tor; in particularly, it focuses on the quality of the anonymity provided.

Elahi et al. [17] address the problem of the entry guard rotation in the Tor protocol. They claim that short-term entry guard churn and explicit time-based entry guard rotation significantly degrade Tor clients anonymity. They also propose an approach to improve the entry guard selection procedure based on trust-based scheme, so that clients pick guards according to how much they trust them.

Biryukov et al. [11] present different vulnerabilities of the Tor protocol. They describe attacks to hidden services, namely denying the hidden service, harvesting hidden service descriptors, identifying entry guards and the deanonymization of hidden services. They point out serious problems in the Tor implementation.

Johnson et al. [13] tie together two important questions about Tor anonymity. What if the attacker runs a relay and what if the attacker can watch part of the Internet? They show that Tor faces great risks from traffic correlation, particularly considering an attacker that can monitor a big part of the network.

These three articles paint a clear picture of the security situation in the Tor network. They address problems in the design and in the implementation of the network that produce anonymity flaws.

6. CONCLUSIONS

Botmasters fight researchers and law enforcement everyday, trying to keep their botnets alive. They design botnets aiming to obtain resilience using any possible means. Lately, they are trying to use the Tor network in order to achieve anonymity for their services and keep their C&C channel hidden. However, we showed that P2P botnets using Tor are still vulnerable to the same kind of attacks such as crawling and sinkholing. Moreover, centralised Tor-based botnets are subject to the vulnerability of Tor itself. In fact, Tor can be affected causing the loss of anonymity if the attacker infects particular relays or a big part of the relays in the network. Seeking more resilient and stealthier properties for the botnets, the botmasters may decide to use Tor assuming that it can provide such properties for free. Instead, botnets are eventually affected by the same anonymity issues that afflict Tor. Hence, even using Tor, the security of the C&C servers is once again compromised.

This does not mean that Tor is not a good solution for botnets but botmasters have to design them taking in consideration the Tor infrastructure and in particular its vulnerabilities. They cannot just exploit the network risking to disrupt it. Instead they have to use it and to enhance it at the same time. Moreover, even at the client side, more accurate techniques can be applied to hide the Tor client. For example, botwriters can compile the Tor source code along with the malware code or apply process hollowing techniques. Tor is an appealing platform for botnets, but the risk of such a platform cannot be underestimated.

REFERENCES:

[1] C. Funk and D. Maslennikov, *IT Threat Evolution Q2 2013*, http://www.securelist.com/en/analysis/204792299/IT, Kaspesky Lab.

[2] T. Dirro, P. Greve, H. Li, F. Paget, V. Pogulievsky, C. Schmugar, J. Shah, R. Sherstobitoff, D. Sommer, B. Sun, A. Wosotowsky, and C. Xu, *McAfee Threats Report: Second Quarter 2013, http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2013.pdf*. McAfee Lab.

[3] J. Salo, *Recent Attacks On Tor*, Aalto University, PhD thesis, 2010.

[4] D. Brown, *Resilient Botnet Command and Control with Tor, http://www.defcon.org/images/defcon-18/dc-18-presentations/D.Brown/DEFCON-18-Brown-TorCnC.pdf*, DefCon 18, 2010

[5] C. Guarnieri, *Skynet, a Tor-powered botnet straight from Reddit, https://community.rapid7.com/community/infosec/blog/2012/12/06/skynet-a-tor-powered-botnet-straight- from-reddit*, Rapid7 2012
[6] R. Dingledine, *Many more Tor users in the past week?, https://lists.torproject.org/pipermail/tor-talk/2013-August/029582.html*, Tor Mailing List, August 2013.
[7] L. Munson, *Tor usage doubles in August. New privacy-seeking users or botnet?*, http://nakedsecurity.sophos.com/2013/08/29/tor-usage-doubles-in-august-new-privacy-seeking-users-or-botnet/29August 2013.
[8] Y. Klijnsma, *Large botnet cause of recent Tor network overload, http://blog.fox-it.com/2013/09/05/large-botnet-cause-of-recent-tor-network-overload/* Fox-It, 5 September 2013.
[9] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, *Highly Resilient Peer-to-Peer Botnets Are Here: An Analysis of Gameover Zeus*, 8th IEEE International Conference on Malicious and Unwanted Software, MALWARE 2013, Fajardo, Puerto Rico, USA.
[10] M. V. Barbera, V. P. Kemerlis, V. Pappas and A. D. Keromytis, *CellFlood: Attacking Tor Onion Routers on the Cheap*, 18th European Symposium on Research in Computer Security (ESORICS). Egham, UK, September 2013.
[11] A. Biryukov, I. Pustogarov, R.P. Weinmann, *Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization*, IEEE Symposium on Security and Privacy 2013.
[12] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, *Denial of Service or Denial of Security? How Attacks on Reliability can Compromise Anonymity*, In the Proceedings of CCS 2007, October 2007.
[13] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, *Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries*, In the Proceedings of the 20th ACM conference on Computer and Communications Security (CCS 2013), November 2013.
[14] *http://www.theguardian.com/world/the-nsa-files* The Guardian.
[15] M.Mimoso, *http://threatpost.com/moving-to-tor-a-bad-move-for-massive-botnet/102284*, The Kaspersky Lab Security News Service.
[16] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich, H. Bos, *SoK: P2PWNED Modeling and Evaluating the Resilience of Peer-to-Peer Botnets*, 34th IEEE Symposium on Security and Privacy, S&P 2013, San Francisco, CA.
[17] T. Elahi, K. Bauer, M. AlSabah, R. Dingledine, I. Goldberg, *Changing of the Guards: A Framework for Understanding and Improving Entry Guard Selection in Tor*, In the Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2012), Raleigh, NC, USA, October 2012.
[18] C. Rossow, *Using Malware Analysis to Evaluate Botnet Resilience*, Phd Thesis, 2013.
[19] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, *Your Botnet is My Botnet: Analysis of a Botnet Takeover*, In Proceedings of the 16th ACM conference on Computer and communications security, CCS09, pages 635647, New York, NY, USA. ACM. 2009.
[20] R. Dingledine, N. Mathewson, P. Syverson, *Tor: The Second-Generation Onion Router*, USENIX Security, 2004.
[21] L. Verlier and P. Syverson, *Locating Hidden Servers*, In the Proceedings of the 2006 IEEE Symposium on Security and Privacy, May 2006.