Detection of Bots and Bot Master in P2P Network- A Survey

[1]Jyothi Belgoankar, [2]Kantharaju H.C
[1]PG Scholar, [2] Assistant Professor
Department of Computer Science and Engineering
Vemana IT, Visvesvaraya Technological University, Belgaum, Karnataka, India.

Abstract **- Peer-to-peer (P2P) botnets have recently been adopted by botmasters for their resiliency against take-down efforts. Modern botnets are sneaky as they perform maliciously, making current detection approaches difficult. The main goal of Bot Master is to increase the network traffic and make the network transaction delay, which in turn affects the user transactions. In addition, the rapidly growing volume of network traffic calls for high scalability of systems. The proposed system i.e., botnet detection system is capable of detecting stealthy P2P botnets in the network using clustering fingerprint concept. Here, itidentifies all hosts or peers that are likely engaged in P2P communications. It derives fingerprints to profile P2P traffic and further distinguish P2P botnet traffic from legitimate traffic. Parallel computation with bound complexity makes scalability a built-in feature of the system. Pervasive evaluation has demonstrated both high detection accuracy and great scalability of the proposed system.**

I.INTRODUCTION

A BOTNET is a collection of compromised hosts (i.e., bots) that are controlled by an attacker (the botmaster) through a command and control (C&C) channel. Botnets serve as the infrastructures responsible for a variety of cyber-crimes, such as spamming, distributed denial- of-service (DDoS) attacks,identity theft etc., The Command and control channel is a very important component of a botnet because botmasters rely on the C&C channel to issue commands to their bots and receive information from the compromised machines. Botnets may structure their C&C channels in different ways.

In a centralized architecture, all bots in a botnet contact one (or a few) C&C server(s) owned by the botmaster. However, a fundamental disadvantage of centralized C&C servers is that they represent a single point of failure. In order to overcome this problem, botmasters have recently started to build botnets with a more resilient C&C architecture, using a P2P structure[1]–[3] or hybrid P2P/centralized C&C structures[4].

Bots belonging to a P2P botnet form an overlay network in which any of the nodes (i.e., any of the bots) can be used by the botmaster to distribute commands to the other peers or collect information from the other peers. Some notable examples of P2P botnets are represented by Nugache[5], Storm[2] , Waledac[4] , and even Confiker, that isshown to embed P2P capabilities[3]. Waledac and Storm are of particular interest because they use P2P C&C structures as the primary way to organize the bots. More complex, and perhaps more costly to manage compared to centralized botnets, P2P botnets offer higher resiliency against take down efforts(e.g., by law enforcement), since even if a significant portion of bots in a P2P botnet are disrupted the remaining bots may still be able to communicate with each other and with the botmaster. However, designing an effective P2P botnet detection system is faced with several challenges. Firstly, the P2P file sharing and communication applications, like Skype,Bittorrent, and emule, are too popular and hence C&C traffic of P2P botnets can easily blend into the backgroundP2P traffic. The challenge is further compounded by the fact that a bot compromised host may exhibit mixed patterns of both legitimate and botnet P2P traffic (e.g., due to the coexistence of a file-sharing P2P application and a P2P bot on the same host).Second, modern botnets possess to use increasingly stealthy ways to perform malicious activities that are extremely hard to be observed in the network traffic. For example, some botnets send spam through large popular webmail services such as Hotmail[6], which is transparent to network detectors due to encryption and overlap with legitimate email use patterns. Third, as the volume of network traffic grow rapidly, the proposeddetection system is required to process a huge amount of information efficiently.

Here, it presents a novel scalable botnet detection system capable of detecting stealthy P2P botnets. It refer to a stealthy P2P botnet as a P2P botnet whose malicious activities may not be observable in the network traffic.Particularly, the system aims to detect stealthy P2P botnet even if P2P botnet traffic is overlapped with traffic generated by legitimate P2P applications (e.g., Skype) running on the same compromised host and Achieve high scalability. To this end, the system identifies P2P bots within a monitored network by detecting the C&C communication patterns that characterize P2P botnets, regardless of how they will perform malicious activities in response to the commands given by botmasters. Specifically, it derives the statistical fingerprints of the P2P communications generated by P2P hosts and leverages them to distinguish between hosts that are part of legitimate P2P networks (e.g., file sharing networks) and P2P bots. The high scalability of the system stems from the parallelized computation with bounded computational

complexity. To summarize, this work makes the following contributions:

1) A new clustering-based analysis approach to identify hosts that engage in P2P communications.
2) An suitable algorithm for P2P traffic profiling, where it builds statistical fingerprints to profile various P2P applications and estimates their active time.
3) A P2P botnet detection method that can effectively detect stealthy P2P bots even if the P2P botnet traffic is overlapped with traffic generated by legitimate P2P applications (e.g., Skype) running on the same compromised machine.
4) A scalable design based on an efficient detection algorithm and parallelized computation.
5) A prototype system and extensive evaluation based on real-world network traffic, which has demonstrated high detection accuracy (i.e., a detection rate of 100% and 0.2% false positive rate) and great scalability (i.e., processing 80 million flows in 0.8 hour) of the design.

II.RELATED WORK

A.Measurements and Mitigation of Peer-to-Peer-based Botnets

Botnets, i.e., networks of compromised machines under a common control infrastructure, it is commonly controlled by an attacker with the help of a central server: all compromised machines connect to the central server and wait for commands. However, the first botnets that use peer-to-peer (P2P) networks for remote control of the compromised machines appeared in the wild recently. Here a methodology is been used to analyze and mitigate P2P botnets. In a case study, it's examined in detail the Storm Worm botnet, the most wide-spread P2P botnet is currently propagating in the wild. It's able to infiltrate and analyze the botnet in depth, it allows us to estimate the total number of compromised machines. Furthermore, it present two different ways to disrupt the communication channel between controller and compromised machines in order to mitigate the botnet and evaluate the effectiveness of these mechanisms.

B.A BotMiner- Clustering Analysis of Network Traffic for Protocol and Structure Independent Botnet Detection.

Botnets are now the key platform for many cyber attacks, such as spam, distributed denial-of-service (DDoS), identity theft, and phishing. Most of the current botnet detection approaches work only on specific botnet command and control (C&C) protocols (e.g., IRC) and structures (e.g., centralized), and become ineffective when botnets change their C&C techniques. Here it presents a general detection framework that is independent of botnet C&C protocol and structure, and requires a priori knowledge of botnets (such as captured bot binaries and hence the botnet signatures, and C&C server names/addresses). It starts from the definition and essential properties of botnets. Botnet is defined as a coordinated group of malware instances that are controlled via C&C communication channels. Most essential properties of a botnet are that the bots communicate with some C&C servers/peers, they perform malicious activities, and do it in a similar or correlated way. Accordingly, the detection framework clusters similar communication traffic and the malicious traffics, and perform cross cluster correlation in order to identify the hosts that share both similar communication patterns and similar malicious activity patterns. These hosts are bots in the monitored network. BotMiner prototype system is implemented and evaluated by using many real network traces. The results shows that it can detect real-world botnets (IRC-based, HTTP-based, and P2P botnets including Nugache and Storm worm), and has too low false positive rate.

C.BotGraph: Large Scale Spamming Botnet Detection

Network security applications often require analyzing huge volumes of data to identify abnormal patterns or activities. The rise of cloud-computing models opens up new opportunities to address this challenge by leveraging the power of parallel computing. Here its designed and implemented a novel system called BotGraph to detect a new type of botnet spamming attacks targeting major Web email providers. BotGraph open up the correlations among botnet activities by constructing large user-user graphs and looking for tightly connected subgraph components. This enables to identify stealthy botnet users that are hard to detect when viewed in isolation. To deal with the huge data volume, BotGraph is implemented as a distributed application on a computer cluster, and discover a huge number of performance optimization techniques. By Applying it to two months of Hotmail log containing over 500 million users, BotGraph would successfully identify over 26 million botnet-created user accounts with a low false positive rate. The run time of constructing and analysing a 220GB Hotmail log is around 1.5 hours with 240 machines. It's believed that both the graph-based approach and the implementations are generally applicable to a wide class of security applications for analysing large datasets.

D.BotGrep -Finding P2P Bots with Structured Graph Analysis

A key feature that distinguishes modern botnets from earlier counterparts is their increasing use of structured overlay topologies. This allows to carry out sophisticated coordinated activities while being resilient to churn, but can also be used as a point of detection. Here, devised techniques is to localize botnet members based on the unique communication patterns arising from their overlay topologies used for C&C. Experimental results on synthetic topologies embedded within Internet traffic traces from an ISP's backbone network indicate that the techniques
(i)It can localize the majority of bots with low false positive count.
(ii) They are resilient to incomplete visibility arising from partial deployment of monitoring systems and

measurement inaccuracies from dynamics of background traffic.

E.Detecting P2P botnets through network behavior analysis and machine learning

Botnets have become one of the major threats on the Internet for serving as a vector for carrying attacks against organizations and committing internet crimes. They are used to create spam, carry out DDOS attacks and click-fraud, and steal sensitive information. Here, it proposes a new approach for characterizing and detecting botnets using network traffic behaviours. This approach focuses on detecting the bots before they launch their attack. The focus is on detecting P2P bots that represent the latest and most challenging types of botnets currently available. The ability of five different commonly used machine learning techniques to meet online botnet detection requirements, such as adaptability, novelty detection, and early detection are studied here. The results of the experimental evaluation based on existing datasets show that it is possible to detect effectively botnets during the botnet Command-and-Control (C&C) phase and before they launch their attacks using traffic behaviours only.

F. Peer to Peer Botnet Detection Using Data Mining Scheme

Botnet was composed of the virus-infected computers severely threaten the security of Network. Hackers, first, inject virus in computers, which will be guided and controlled by them via the internet to operate distributed denial of services (DDoS), hack private information, share unwanted mails and other malicious activities. By counterfeiting P2P software, P2P botnet used many main controller to avoid single point failure, and fails many misuse detecting systems together with encryption technologies. Differentiating it from the normal network behavior, P2P botnet sets many sessions without consuming bandwidth substantially, making itself exposed to the anomaly detection system. The mining scheme was verified in internet to prove its capability of discovering the host of P2P botnet. Essentially, the analysis applied the original dissimilarity of P2P botnet differing from normal internet behaviors as parameters of data mining, which were then grouped and distinguished to obtain reliable results with acceptable accuracy.

III.SYSTEM DESIGN

A P2P botnet relies on a P2P protocol to establish a C&C channel and communicate with the botmaster. Therefore P2P bots exhibit some network traffic patterns that are common to other P2P client applications (either legitimate or malicious). Thus, the system is divided into two phases. In first phase, aim is at detecting all hosts within the monitored network that engage in P2P communications.As shown in figure 1,Analyze raw traffic collected at the edge of the monitored network and apply a pre-filtering step to discard network flows that are unlikely to be generated by P2Papplications. Then analyze the remaining traffic and extract a number of statistical features to identify flows generated by P2P clients. In second phase, system analyses the traffic generated by the P2P clients and classifies them into either legitimate P2P clients or P2P bots. Specifically, Investigate the active time of a P2P client and identify it as a candidate P2P bot if it is persistently active on the underlying host. Further analyze the overlap of peers contacted by two candidate P2P bots to finalize detection.

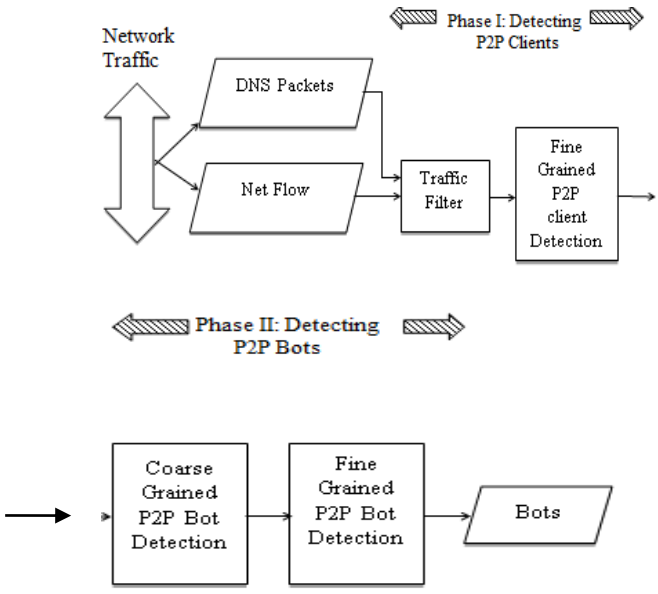

Fig 1: System Overview

A. Identifying P2P Clients

The main aim is at filtering the network traffic that is not related to P2P communications. It is accomplished by passively analyzing DNS traffic, and identifying the network flows whose destination IP addresses were previously resolved in DNS responses. Specifically, it leverage the following feature: P2P clients usually contact their peers directly by looking up IPs from a routing table for the overlay network, rather than resolving a domain name.Since most non-P2P applications (e.g., browsers, email clients, etc.) often connect to a destination address resulting from domain name resolution, this simple filter can eliminate a very large percentage of non-P2P traffic, while retaining the vast majority of P2P communications.

Fine Grained Detection of P2P Clients: This component is responsible for filtering the P2P clients by analyzing the remaining network flows after the P2P Traffic Filter component. For each host 'h' within the monitored network we identify two flow sets, denoted as Stcp(h) and Sudp(h), which contain the flows related to successful outgoing TCP and UDP connection, respectively. It is considered as successful, for those TCP connections with a completed SYN, SYN/ACK, ACK handshake, and for UDP (virtual) connections which has at least one "request" packet and a consequent response packet. In order to filter P2P clients, first consider the fact that each P2P client frequently

exchanges control messages (e.g., ping/pong messages) with other peers. The characteristics of these messages, such as the size and frequency of the exchanged packets, are similar for nodes in the same P2P network, and vary depending on the P2P protocol and network in use.

To identify flows corresponding to P2P control messages, we first apply a flow clustering process intended to group together similar flows for each candidate P2P node 'h'. Given sets of flows Stcp(h) and Sudp(h), characterize each flow using a vector of statistical features v(h) = [Pkts, Pktr, Bytes, Byter], in whichPkts and Pktr represent the number of packets sent and received, and Bytes and Byter represent the number of bytes sent and received, respectively.The distance between two flows is defined as the euclidean distance of their two corresponding vectors. Then apply a K mean clustering algorithm to partition the set of flows into a number of clusters based upon their size (Bytes sent + received). Each of the obtained clusters of flows, Cj(h), represents a group of flows with similar size. Take a cluster and check for distinct BGP prefix for each destination IP address if BGP prefix count is lesser then threshold values then discard those clusters. Remaining vectors will be called fingerprint clusters.

B. Detecting P2P Bots

Coarse-Grained Detection of P2P Bots: Since bots are malicious programs used to perform profitable malicious activities, they represent valuable assets for the botmaster, who will try to maximize utilization of bots. This is particularly true for P2P bots because in order to have a functional overlay network (the botnet), a sufficient number of peers needs to be always online. Hence, it aims at identifying P2P clients that are active for a time TP2P close to the active time Tsys of the underlying system they are running on. While this behavior is not unique to P2P bots and may be representative of other P2P applications (e.g., Skype clients that run for as long as a machine is on). To estimate Tsys we proceed as follows. For each host h ∈ H that are identified as P2P clients consider the timestamp tstart(h) of the first network flow observed from h and the timestamp tend(h) related to the last flow thats seen from h. Afterwards, we divide the time tend(h)−tstart(h) into w epochs (e.g., of one hour each), denoted as T =[t_1,...t_i,...,t_w]. We further compute a vector A(h,T) =[a_1,...a_i,...,a_w] where a_i is equal to 1 if h generated any network traffic between t_{i-1} and t_i. It then estimate the active time of h as Tsys =$\sum^{w}_{i=1}a_i$. In order to estimate the active time of a P2P application, it can leverage obtained fingerprint clusters. It is because that a P2P application periodically exchanges network control (e.g., ping/pong) messages with other peers as long as the P2P application is active. For each host h, examine the set of its fingerprint clusters FC(h) ={FC1,...FCj ...,FCk}. Based on the flows belonging to a fingerprint cluster FCj, use the same approach of computing Tsys to calculate its active time, denoted as T(FCj). Then find r(h)=T_{p2p}/T_{sys} and if it is grater then a threshold value say 0.5 then add it as a candidate for p2p bot.

Fine Grained Detection of P2P Bots: The aim is to identify P2P bots from all persistent P2P clients (i.e., set P). It leverage one feature: the overlap of peers contacted by two P2P bots belonging to the same P2P botnet is much larger than that contacted by two clients in the same legitimate P2P network.If two P2P clients (say ha and hb) belong to the same P2P network, regardless of a legitimate P2P network or a P2P botnet network, these two clients will follow the same implementation of the identical P2P protocol. Hence, the network flows corresponding to the same type of P2P control messages (e.g., ping/pong messages) will exhibit similar flow sizes across P2P clients running the same P2P application. Since a fingerprint cluster summarizes network flows for the same type of control messages in one client, two fingerprint clusters corresponding to the same P2P control messages belonging to the same P2P application will have similar flow size. In other words, two P2P clients from the same P2P network will share at least one pair of fingerprint clusters, which have a small value of dbytes(FC(a) i , FC(b) j) since they are corresponding to the same P2P control message. Otherwise, if two P2P clients belong to different P2P networks, dbytes tends to be large. Given two P2P bots (say ha and hb) belonging to the same botnet, the sets of peers contacted by these two bots,will share a large overlap, thereby generating a small value of dIPs(FC(a) i , FC(b) j). Otherwise, if two P2P clients belong to i) the same legitimate P2P network or ii) different P2P networks, they will share a small overlap and produce a large value of dIPs(FC(a) i , FC(b) j). It further defines a distance function dist(ha,hb) to quantify the similarity of two P2P clients by integrating dbytes and dIPs. dist(ha,hb) tends to yield a small value if ha and hb are infected with bots from the same P2P botnet. Especially, even if ha and hb are infected with P2P bots from the same botnet and they run legitimate P2P applications simultaneously, the distance quantified by dist(ha,hb) will be small. It is because that at least one pair of fingerprint clusters that are generated by P2P bots will yield small values for both dbytes and dIPs.

CONCLUSION

Here, it presents a novel scalable botnet detection system that is able to detect stealthy P2P botnets, whose malicious activities may not be observable in the network traffic. To accomplish it, statistical fingerprints of the P2P communications to first detect P2P clients and distinguish between those that are part of legitimate P2P networks and P2P bots.

REFERENCES

[1] Argus: Auditing Network Activity [Online]. Available: http://www.qosient.com/argus/
[2] Autoit Script [Online]. Available: http://www.autoitscript.com/autoit3/index.shtml
[3] (2011). Zeus Gets More Sophisticated Using P2P Techniques [Online]. Available: http://www.abuse.ch/?p=3499
[4] A. Binzenhofer, D. Staehle, and R. Henjes, "On the stability of chord- based P2P systems," in Proc. IEEE Global Telecommun. Conf., vol. 2. Nov./Dec. 2005, pp. 884–888.

[5] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in Proc. ACM SIGMETRICS, 2005, pp. 50–60.
[6] D. Dagon, G. Gu, C. Lee, and W. Lee, "A taxonomy of botnet structures," in Proc. 33rd Annu. Comput. Security Appl. Conf., 2007, pp. 325–339.
[7] D. Liu, Y. Li, Y. Hu, and Z. Liang, "A P2P-botnet detection model and algorithms based on network streams analysis," in Proc. IEEE FITME, Oct. 2010, pp. 55–58.
[8] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to- peer networks," in Proc. 6th ACM SIGCOMM Conf. IMC, 2006, pp. 189–202.
[9] G. Bartlett, J. Heidemann, C. Papadopoulos, and J. Pepin, "Estimating P2P traffic volume at USC," USC/Information Sciences Institute, Los Angeles, CA, USA, Tech. Rep. ISI-TR-2007-645, 2007.
[10] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in Proc. USENIX Security, 2008, pp. 139–154.
[11] G. Sinclair, C. Nunnery, and B. B. Kang, "The waledac protocol: The how and why," in Proc. 4th Int. Conf. Malicious Unwanted Softw., Oct. 2009, pp. 69–77.
[12] J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, and X. Luo, "Detecting stealthy P2P botnets using statistical traffic fingerprints," in Proc. IEEE/IFIP 41st Int. Conf. DSN, Jun. 2011, pp. 121–132.
[13] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster, "Boosting the scalability of botnet detection using adaptive traffic sampling," in Proc. 6th ACM Symp. Inf., Comput. Commun. Security, 2011, pp. 124–134.
[14] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," J. Intell. Inf. Syst., vol. 17, nos. 2–3, pp. 107–145, 2001.
[15] M. P. Collins and M. K. Reiter, "Finding peer-to-peer file shar- ing using coarse network behaviors," in Proc. 11th ESORICS, 2006, pp. 1–17.
[16] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the storm (peacomm) worm," Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep., 2007.
[17] P. Porras, H. Saidi, and V. Yegneswaran. (2009). Conficker C Analysis [Online]. Available: http://mtc.sri.com/Conficker/addendumC/index.html
[18] Resilient Botnet Command and Control with Tor [Online]. Available: http://www.defcon.org/images/defcon-18/dc-18-presentations/D.Brown/DEFCON-1%8-Brown-TorCnC.pdf 2010
[19] R. Lemos. (2006). Bot Software Looks to Improve Peerage [Online]. Available: http://www.securityfocus.com/news/11390
[20] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in Proc. USENIX Security, 2010, pp. 1–16.
[21] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a DHT," in Proc. Annu. Conf. USENIX Annu. Tech. Conf., 2004, pp. 127–140.
[22] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, et al., "Detecting P2P botnets through network behavior analysis and machine learning," in Proc. 9th Annu. Int. Conf. PST, Jul. 2011, pp. 174–180.
[23] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in Proc. 13th ACM Int. Conf. WWW, 2004, pp. 512–521.
[24] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugachetrojans: P2P is here," in Proc. USENIX, vol. 32. 2007, pp. 18–27.
[25] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm," in Proc. USENIX LEET, 2008, pp. 1–9.
[26] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in Proc. 4th ACM SIGCOMM Conf. IMC, 2004, pp. 121–134.
[27] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in Proc. ACM SIGCOMM, 2005, pp. 229–240.
[28] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in Proc. ACM SIGMOD, 1996, pp. 103–114.
[29] T.-F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in Proc. ICDCS, Jun. 2010, pp. 241–252.
[30] W. Liao and C. Chang, "Peer to peer botnet detection using data mining scheme," in Proc. IEEE Int. Conf. ITA, Aug. 2010, pp. 1–4.
[31] Y. Zhao, Y. Xie, F. Yu, Q. Ke, and Y. Yu, "Botgraph: Large scale spamming botnet detection," in Proc. 6th USENIX NSDI, 2009, pp. 1–14
[32] Z. Li, A. Goyal, Y. Chen, and A. Kuzmanovic, "Measurement and diagnosis of address misconfigured P2P traffic," in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.