A Survey of Botnet Technology and Defenses

Michael Bailey, Evan Cooke, Farnam Jahanian, Yunjing Xu
University of Michigan
Ann Arbor, Michigan
{mibailey, emcooke, farnam, yunjing}@umich.edu

Manish Karir
Merit Network, Inc.
Ann Arbor, Michigan
mkarir@merit.edu

Abstract

Global Internet threats have undergone a profound transformation from attacks designed solely to disable infrastructure to those that also target people and organizations. At the center of many of these attacks are collections of compromised computers, or Botnets*, remotely controlled by the attackers, and whose members are located in homes, schools, businesses, and governments around the world [6]. In this survey paper we provide a brief look at how existing botnet research, the evolution and future of botnets, as well as the goals and visibility of today's networks intersect to inform the field of botnet technology and defense.*

1 Introduction

Global Internet threats are undergoing a profound transformation from attacks designed solely to disable infrastructure to those that also target people and organizations. This alarming new class of attacks directly impacts the day-to-day lives of millions of people and endangers businesses and governments around the world. For example, computer users are assailed with spyware that snoops on confidential information, spam that floods email accounts, and phishing scams that steal identities.

At the center of many of these attacks is a large pool of compromised computers located in homes, schools, businesses, and governments around the world. Attackers use these *zombies* as anonymous proxies to hide their real identities and amplify their attacks. *Bot* software enables an operator to remotely control each system and group them together to form what is commonly referred to as a *zombie army* or *botnet* [6]. The scope of the botnet problem is difficult to quantify, as the highly covert nature of bots and botnets makes them difficult to identify and even harder to measure. Nevertheless, CERT has identified botnets with more than 100,000 members, and almost 1 million bot infected hosts have been reported [19].

In this paper, we provide a survey of current botnet technology and defense by exploring the intersection between existing botnet research, the evolution of botnets themselves, and the goals and perspectives of various types of networks. In section 2, we provide a brief overview of botnets to highlight the invariant nature of their behavior in various phases of their life-cycle. Then, in section 3, we describe how different kinds of networks have access to different types of visibility and this has a strong impact on the effectiveness of any botnet detection mechanism. Next, in section 4, we provide a comprehensive overview of the various botnet detection techniques that have been been proposed. Finally, in section 5, we summarize our survey and suggest future directions.

2 Understanding Botnets

In many respects, the bots found in the wild today are a hybrid of previous threats. They can propagate like worms, hide from detection like many viruses, attack like many stand-alone tools, and have an integrated command and control system. Even more concerning, the construction of bots is now very much a cooperative effort. An example is the source code of SDBot, which contains comments from many different authors. The result is a proliferation of different bot variants. A recent Microsoft survey found more than 43,000 new variants of backdoor trojans and bots during the first half of 2006 [20].

2.1 Propagation and Compromise

One core problem for botnet attackers is how to get bots onto victim computers. Because very few users would actually agree to have their computers used to conduct packet floods, attackers surreptitiously install their malicious software. This process of getting malicious software on victim's hosts has evolved significantly over time. One change that happened a few years ago is the shift from a single propagation vector, that might have required a manual installation process by the attacker, to multiple automated propagation vectors. For example, The Slammer worm used a single

978-0-7695-3568-5/09 $25.00 © 2009 IEEE
DOI 10.1109/CATCH.2009.40

Table 1. Propagation Mechanisms

Propagation Methodology		Design Complexity	Detectability	Propagation Speed	Population Size
Exploit:	Operating System	*Medium*	*High*	*Low*	*High*
	Services	*Medium*	*Medium*	*Medium*	*Medium*
	Applications	*High*	*Low*	*High*	*Low*
Social Engineering		*Low*	*Medium*	*Low*	*High*

Table 2. Command and Control Topologies

Topology	Design Complexity	Detect-ability	Message Latency	Surviv-ability
Centralized	*Low*	*Medium*	*Low*	*Low*
Peer-to-Peer	*Medium*	*Low*	*Medium*	*Medium*
Unstructured	*Low*	*High*	*High*	*High*

Table 3. Attack Classes

Topology	Detectability	Design Complexity	Attack Value
Single Host DDoS	*High*	*Low*	*Low*
Multi Host DDoS	*Medium*	*Medium*	*Medium*
Identity Theft	*Low*	*High*	*Medium*
Spam	*Medium*	*Medium*	*High*
Phishing	*Medium*	*High*	*Medium*

vulnerability to infect hosts while more modern bots have many distinct, completely automated propagation vectors. For example, SDBot (also known as rBot) propagates using a number of different mechanisms including open files shares, P2P networks, backdoors left by previous worms, and using exploits of numerous common Windows vulnerabilities.

Another important shift in propagation behavior is the move away from random scanning to robust "hitlists" (e.g., lists of hosts, email lists, social networking lists - buddy list in AIM, ARP cache entries, etc), from vulnerable services, to vulnerable applications, to "vulnerable" users (or social engineering). Table 2.1 illustrates this evolution. Some of the very first self propagating software, such as the Morris worm, exploited operating systems or low-level services to gain entry into a system. Since then, there has been a steady shift up toward targeting higher-level applications like web browsers and social engineering attacks against users. For example, drive-by downloads and web-based infection vectors are now commonplace with a recent google study showing hundreds of thousands of malicious URLs exploiting software such as Flash Player and installing trojans, adware, and other malicious code [21]

2.2 Command and Control

A second core problem for botnet attackers is how to communicate with each bot instance. Most attackers would like the ability to rapidly send instructions to bots but also do not want that communication to be detected or the source of the those commands to be revealed. To explore the implications of various bot communication methods, we identify three possible topologies and investigate their associated benefits and weaknesses as shown in Table 2.2.

Centralized: A centralized topology is characterized by a central point that forwards messages between clients. Messages sent in a centralized system tend to have low latency as they only need to transit a few well-known hops. From the perspective of an attacker, centralized systems have two major weaknesses: they can be easier to detect since many clients connect the same point, and the discovery of the central location can compromise the whole system.

P2P: Peer-to-peer (P2P) botnet communication has several important advantages over centralized networks. First, a P2P communication system is much harder to disrupt. This means that the compromise of a single bot does not necessarily mean the loss of the entire botnet. However, the design of P2P systems are more complex and there are typically no guarantees on message delivery or latency.

Unstructured: A botnet communication system could also take the P2P concept to the extreme and be based on the principle that no single bot would know about any more than one other bot. In such, a topology a bot or controller that wanted to send a message would encrypt it and then randomly scan the Internet and pass along the message when it detected another bot. The design of such a system would be relatively simple and the detection of a single bot would never compromise the full botnet. However, the message latency would be extremely high, with no guarantee of delivery.

In practice, Botnet communication has become steadily more sophisticated—moving from simple readily detectable IRC communication to complex anonymity providing P2P communication. An excellent modern example is the Nugache botnet, which emerged in 2006, and has a true peer-to-peer structure that is highly resilient to disruption or takeover. As a result, the existence of large botnets based on this technology have long escaped public attention [9].

2.3 Attacks and Theft

The third core problem for botnet attackers is how to extract value from a bot infected node. In the past, this value might have been a denial of service (DoS) attack to punish another IRC user or gain status and reputation in the underground community. Attackers have since found new ways to create value and even extract real monetary gain as shown in Table 2.3.

Botnets used to initiate simple DoS attacks quickly evolved into multi-host distributed DDoS attacks involving large numbers of computers. SDBot and Agobot both have remotely accessible commands for initiating DDoS attacks. Such capabilities were used in DDoS extortion scams that provided attackers with real financial gain.

Attackers also discovered that there is value in the information stored on infected computers and on the networks in which they are positioned. Attackers can use stolen credit card, social security numbers, and other personal information for identity theft and to commit industrial espionage. One example of a botnet that uses advanced key logging techniques to collect personal information is SDBot. Variants of SDBot look for passwords such as Paypal accounts and some will install generic keylogging tools such as carnivore.

However, one of most important use of bots is to send Spam. Sending Spam requires large numbers of new mail servers (as the old ones get blocked) and bot-infected hosts proved to be the perfect tool. For example, the Storm botnet has a remotely controllable interface for conducting Spam campaigns and a large number of hosts in the Storm botnet were used to send millions of Spam messages.

Finally, botnets are also used a flexible platforms from which to run arbitrary network services such as for phishing attacks. Attackers can extract value from bots by turning them into web servers or DNS servers to conduct phish attacks and other identity theft scams.

3 Understanding Networks

Botnets and the techniques proposed to detect and mitigate them do not exist in a vacuum, they must be deployed to be effective. In this section, we discuss the goals of various networks and explore the issues of data sources and visibility as they relate to botnet detection and mitigation.

3.1 Differing Organizations and Goals

Networks can be broadly placed into two categories: service provider networks and enterprise networks. While much of the infrastructure and basic principles of networking and security apply to both, the goals of these organizations are oftentimes different. Network security at a service provider is mostly concerned with ensuring the survivability of the network services and preventing abuse and network security at an enterprise is mostly concerned with operating and maintaining a secure computing environment. As a result, the enterprise network is concerned with cleaning up infected hosts and preventing the spread of compromised machines while the network service provider is focused on notification of malicious activity to the customers with sufficient information to help them track compromised hosts.

3.2 Data Sources and Botnet Detection

One of the most important aspects of this distinction between different types of organizations is the different data types that are available. An enterprise network might have access to DHCP logs, DNS resolver data, address allocation data, complete packet traces for each host, email server logs, policy data, as well as antivirus scanning logs. A network service provider on the other hand might only have access to sampled or unsampled netflow data and perhaps some limited packet tap data. While it is possible to infer activity such as DNS requests or SMTP activity, the accuracy and confidence in this data would depend on the netflow sampling being used. Consider, for example, several of the prevalent data types below:

- **DNS Data:** Data regarding name resolution can be obtained by mirroring data to and from the local DNS servers or resolvers and can be used to detect both botnet attack behavior such as email spam (MX query lookups), as well as botnet communication behavior such as DNS lookups for suspicious domains.
- **Netflow Data:** Netflow data represents information gathered from the network by sampling traffic flows and obtaining information regarding source and destination IP addresses and port numbers. At a course level, this data is useful for identifying malicious communication patterns and course grained attacks, but often visibility is limited to the peering edge of a network, missing large amounts of backbone (ISP) or switched (enterprise) traffic.
- **Packet Tap Data:** Packet tap data, while providing a more fine grained view than netflow and offering an attractive deployment model (switches or taps, not routers), is generally more costly in terms of hardware and computation. While providing a much deeper level of insight for signature-based detection algorithms, simple encryption reduces this visibility back to the same order as netflow.
- **Address Allocation Data:** Knowing where hosts and users are in the network can be a powerful tool for identifying reconnaissance behaviors of bots and for

tying them to specific machines or users. Internal routing protocols, such as OSPF, and dynamic allocation protocols, such as DHCP, provide a level of detail generally unavailable to the bots, but this visibility is generally reserved for the enterprise only.

- **Honeypot Data:** The use of sacrificial hosts, placed in the network with the express intention of them being turned into bot members, can be a powerful tool for gaining insight into botnet means and motives without actually involving production hosts. Unfortunately, as propagation techniques tend towards social engineering, these honeypots must increasingly emulate not only user systems but the users themselves to be useful.

- **Host Data:** Host level data, from operating system and application configurations, antivirus and firewall logs, to user activity (e.g., attaching a process name to a network flow), provides a wealth of security information and can avoid the visibility issues with encrypted data. Unfortunately, visibility into these behaviors are limited to the network edge, and this often requires instrumenting tens of thousands of devices.

4 Understanding Techniques

In this section, we survey some of the existing work in detecting and understanding botnets. While a complete survey is not possible in such limited space, we find that current research on botnets falls roughly into two broad categories—botnet detection techniques and botnet measurement studies.

4.1 Detection Techniques

Detection via cooperative behaviors Bothunter [14] modeled the bot infection phase as a set of loosely ordered communication flows that are exchanged between an internal host and one or more external entities and used this model to compare suspected infection events. Botsniffer [15] proposed statistical algorithms to detect botnets based on their multiple crowd-like behaviors (e.g. sending spam, scanning and binary downloading) in a centralized topology. Botminner [13] extended botsniffer and proposed a detection framework that performs clustering on monitored C&C communication and malicious activities respectively, then a cross-correlation on them to generate the final detection results. Karasaridis *et al.* designed a detection scheme to calculate the distances between monitored flow data and a pre-defined IRC traffic flow model [18]. Akiyama *et al.* defined three metrics to determine the cooperative behavior of botnets: relationship, response, and synchronization [1]. Strayer *et al.* proposed a temporal correlation algorithm in a five-dimensional space about packet inter-arrival time and packet size [26]. Chois *et al.* observed anomaly group activities of botnets in DNS traffic and used them to do detection [5]. Ramachandram *et al.* discovered identities of bots based on the insight that botmasters themselves must perform "reconnaissance" lookups to determine their bots' blacklist status [24].

Detection by signatures Goebel *et al.* used regular expressions to represent sets of suspicious IRC nick names, and used n-gram analysis and scoring systems to evaluate the nick names to determine if a particular conversation belongs to a bot contaminated host [11]. Binkley *et al.* [3] grouped IP hosts seen in an IRC channel with IPs performing scanning to determine if they were malicious.

Detection of attack behaviors Brodsky *et al.* [4] relied on an assumption that botnets tend to send large number of spam in a relatively short period of time to detect botnet generated spam. Similarly, Xie *et al.* [28] used spam server traffic properties and spam payload to construct a spam signature generation framework.

4.2 Measurement Studies

Measurement studies help defenders better understand the botnet phenomenon and the characteristics of specific types of botnets. Zhu *et al.* created a survey of various areas of Botnet research, including bot anatomy, wide-area measurement studies, botnet modeling and future botnet prediction, honeynet and traffic monitoring [29]. Dagon *et al.* [7] measured three botnets topologies (centralized, peer-to-peer, and random) using three metrics (effectiveness, efficiency, and robustness). In addition to these two general papers, there are many measurement papers with specific emphasis.

Size estimation The majority of botnets measurement papers devote their efforts to estimating the populations of various kinds of botnets in today's Internet. Rajab *et al.* [22] observed the botnet phenomenon from three different perspectives (DNS, IRC, passive). Zhuang *et al.* [30] grouped spam-generating bots into botnets by examining spam contents. Rajab *et al.* [23] considered the discrepancies in botnet size estimation and suggested that botnet size should be a qualified term that is relevant only within the context of the counting method used to generate the result.

Behavior analysis Gianvecchio *et al.* [10] investigated the different statistical patterns of human and irc bot behaviors in a large commercial chat network. Gianvecchio

Table 4. The relationship between the network visibility, the botnet invariant behaviors, and various proposed techniques

		Bot Behaviors		
		Propagation	**Communication**	**Attack**
Data Sources	**Traffic Flows**	scan-detection [14, 15, 13, 3, 18, 26] binary-downloading-detection [14, 15, 13, 26]	control-protocols [14, 15, 13, 11, 3] [18, 1, 26]	ddos-detection [18, 1, 26] spam-detection [15, 13, 18, 4, 28] active-responder [25]
	Darknet Data	bot-informants [14, 13] scan-detection [14, 13]	bot-informants [14, 15, 13]	bot-informants [13]
	Packet Capture	vulnerability-signature [14]	control-signatures [18, 1, 11, 3]	
	DNS Logs		rendezvous-detection [18, 5]	spam-detection [15, 13, 4] reconnaissance-detection [24] active-responder [25]

et al. [10] proposed two types of classifiers (entropy rate and machine learning, respectively) to differentiate human and irc bots. Instead of botnets that send the spam, Anderson *et al.* [2] focused on the scam hosting infrastructure and how it is shared. Dagon *et al.* [8] noted time zones and locations play a critical role in malware propagation.

Peer-to-peer botnets Grizzard et al. [12] provided a history and overview of P2P botnets. Holz et al. [16] presented a case study on Storm including its system-level and network-level behaviors. Kanich et al. [17] tried to present a more accurate estimation for the size of Storm botnet by taking various types of noise (e.g. protocol aliasing, adversarial aliasing, and temporal dynamics) into consideration. Wang et al. [27] summarized the disadvantages of centralized and P2P botnets and proposed a hybrid structured botnet that overcame those disadvantages.

5 Discussion

The previous sections on Understanding Botnets (Section 2), Understanding Networks (Section 3), and Understanding Techniques (Section 4) each highlighted the unique challenges faced by today's botnet technology and defenses. The relationship between these areas can be seen concisely in Table 4 which shows the network visibility, the botnet invariant behaviors, and various proposed techniques and how they intersect. This table and our previous discussion argue:

- **Botnets are moving targets.** All aspects of the botnet's life-cycle, from propagation, to command and control, and attacks are all evolving constantly. Trying to nail down a specific set of tradeoffs (e.g., survivability verses message latency) or predicting future trends is a losing battle.
- **No technique is perfect.** Each detection algorithm or technique comes with its own unique set of tradeoffs with respect to false positives and false negatives and each technique makes a set of assumption about the available insight into the threat and about the aspect of botnet behavior it is discovering.
- **All networks are not the same.** Different types of networks (e.g., enterprises, ISPs) approach the botnet problem with differing goals (i.e., notification verse remediation), with different visibility into the botnet behaviors, and different sources of data with which to uncover those behaviors (e.g., network data, host data).

A successful solution for botnet detection and mitigation will need to cope with each of these realities and their complex interactions with each other.

6 Acknowledgements

This work was supported in part by the U.S. Department of Homeland Security Science & Technology Directorate under Contract No. NBCHC060090.

References

[1] M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi. A proposal of metrics for botnet detection based on its cooperative behavior. In *Proceedings of the 2007 International Symposium on Applications and the Internet Workshops(SAINT-W'07)*, Washington, DC, May 2007.

[2] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In *Proceedings of the 16th USENIX Security Symposium (Security'07)*, Boston, MA, August 2007.

[3] J. R. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06)*, San Jose, CA, July 2006.

[4] A. Broadsky and D. Broksky. A distributed content independent method for spam detection. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[5] H. Choi, H. Lee, H. Lee, and H. Kim. Botnet detection by monitoring group activities in dns traffic. In *Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT'07)*, Washington, DC, October 2007.

[6] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005 Workshop)*, Cambridge, MA, July 2005.

[7] D. Dagon, G. Gu, C. P. Lee, and W. Lee. A taxonomy of botnet structures. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC'07)*, Florida, USA, November 2007.

[8] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proceedings of the 13rd Network and Distributed System Security Symposium (NDSS'06)*, San Diego, CA, February 2006.

[9] D. Dittrich and S. Dietrich. P2p as botnet command and control: a deeper insight. In *Proceedings of the 2008 3rd International Conference on Malicious and Unwanted Software (Malware 2008)*, Alexandria, VA, Oct 2008.

[10] S. Gianvecchio, M. Xie, Z. Wu, and H. Wang. Measurement and classification of huamans and bots in internet chat. In *Proceedings of the 17th USENIX Security Symposium (Security'08)*, San Jose, CA, July 2008.

[11] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[12] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-Peer Botnets: Overview and case study. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[13] G. Gu, R. Perdisci, junjie Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium (Security'08)*, San Jose, CA, July 2008.

[14] G. Gu, P. Porras, V. Yegneswaran, M. Frog, and W. Lee. BotHunter: Detecting malware infection through ids-driven dialog correlation. In *Proceedings of the 16th USENIX Security Symposium (Security'07)*, Boston, MA, August 2007.

[15] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network & Distributed System Security Symposium (NDSS'08)*, San Diego, CA, February 2008.

[16] 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 *First Usenix Workshop on Large-scale Exploits and Emergent Threats (LEET'08)*, San Francisco, CA, April 2008.

[17] C. Kanich, K. Lechenko, B. Enright, G. M. Voelker, and S. Savage. The Heisenbot Uncertainty Problem: Challenges in separating bots from chaff. In *First Usenix Workshop on Large-scale Exploits and Emergent Threats (LEET'08)*, San Francisco, CA, April 2008.

[18] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[19] L. McLaughlin. Bot software spreads, causes new worries. *IEEE Distributed Systems Online*, 5(6), June 2004.

[20] Microsoft. Microsoft security intelligence report: July-december 2006. `http://www.microsoft.com/technet/security/default.mspx`, May 2007.

[21] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser: Analysis of web-based malware. In *First Workshop on Hot Topics in Understanding Botnets, HotBots'07*. USENIX, 2007.

[22] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of Internet Measurement Conference 2006 (IMC'06)*, Rio de Janeiro, Brazil, October 2006.

[23] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My Botnet is Bigger than Yours (Maybe, Better than Yours): why size estimates remain challenging. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[24] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using dnsbl counter-intelligence. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'06)*, San Jose, CA, July 2006.

[25] S. Small, J. Mason, and F. Monrose. To Catch a Predator: A natural language approach for eliciting malicious payloads. In *Proceedings of the 17th USENIX Security Symposium (Security'08)*, San Jose, CA, July 2008.

[26] W. T. Strayey, R. Walsh, C. Livadas, and D. Lapsley. Detecting botnets with tight command and control. In *31st IEEE Conference on Local Computer Networks (LCN06)*, Tampa, Florida, November 2006.

[27] P. Wang, S. Sparks, and C. C. Zou. An advanced hybrid peer-to-peer botnet. In *First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, April 2007.

[28] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming Botnets: Signatures and characteristics. In *Proceedings of ACM SIGCOMM'08*, Seattle, WA, August 2008.

[29] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han. Botnet research survey. In *2008 32nd Annual IEEE International Computer Software and Applications Conference (COMPSAC'08)*, Turku, Finland, July 2008.

[30] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, I. Osipkov, G. Hulten, and J. D. Tygar. Characterizing botnets from email spam records. In *First Usenix Workshop on Large-scale Exploits and Emergent Threats (LEET'08)*, San Francisco, CA, April 2008.