Walowdac – Analysis of a Peer-to-Peer Botnet*

Ben Stock[1], Jan Göbel[1], Markus Engelberth[1], Felix C. Freiling[1], and Thorsten Holz[1,2]

[1]Laboratory for Dependable Distributed Systems
University of Mannheim

[2]Secure Systems Lab
Technical University Vienna

Abstract

A *botnet* is a network of compromised machines under the control of an attacker. Botnets are the driving force behind several misuses on the Internet, for example spam mails or automated identity theft. In this paper, we study the most prevalent peer-to-peer botnet in 2009: *Waledac*. We present our infiltration of the Waledac botnet, which can be seen as the successor of the Storm Worm botnet. To achieve this we implemented a clone of the Waledac bot named *Walowdac*. It implements the communication features of Waledac but does not cause any harm, i.e., no spam emails are sent and no other commands are executed. With the help of this tool we observed a minimum daily population of 55,000 Waledac bots and a total of roughly 390,000 infected machines throughout the world. Furthermore, we gathered internal information about the success rates of spam campaigns and newly introduced features like the theft of credentials from victim machines.

1 Introduction

Botnets, i.e., networks of compromised under the control of an attacker, are nowadays one of the most severe threats in Internet security. With a large number of participating bots, a tremendous number of spam emails can be send out to for example acquire new hosts, or to advertise a diverse set of products. Additionally, Distributed Denial of Service (DDoS) attacks threaten every system connected to the Internet. Therefore, it is crucial to explore ways of detecting and mitigating current and new botnets.

The first generation of botnets uses a central Command & Control (C&C) server to dispatch commands. This type of botnet is rather well-understood [4, 7, 14] and the technique of *botnet tracking* [7] is now a standard method to mitigate this type of threat. The second generation of botnets tries to avoid a centralized infrastructure by using a peer-to-peer-based communication mechanism [8]. The most prominent example for this class of botnets is the Storm Worm botnet that used a distributed hash table as communication medium. This mechanism offered a rather centralized way to infiltrate and analyze the botnet [10, 11].

The Waledac botnet can be regarded as the successor of Storm Worm. However, Waledac uses a more decentralized store-and-forward communication paradigm and new communication protocols so we had to develop novel techniques to track this botnet.

Related Work. The communication protocol used by Waledac was already studied by Leder [12] and Sinclair [15]. Symantec [16] and Trend Micro [1] recently released reports about the implemented features of Waledac, including the spam template system, as well as attempts to measure the size of the botnet. A study by ESET [2] estimated the size of the Waledac botnet around 20,000 bots.

Another study focussing on Waledac was performed by Borup [3]. The focus of this work is on the C&C protocol, the binary obfuscation tech-

*Thorsten Holz was supported by the WOMBAT and FORWARD projects funded by the European Commission, and the *FIT-IT Trust in IT-Systems* (Austria) under the project TRUDIE (P820854).

niques, as well as mitigation methods against the botnet. The author describes a *Sybil attack* [6] that we also used to generate the statistical data that we present in this paper.

Contributions. In this paper, we present the results of yet another analysis of Waledac. Our focus was to try and verify previous measurements as well as building and refining tools to study the botnet efficiently. In contrast to the analysis of previous decentralized botnets, a simple crawling of active peers was no solution to gather in-depth information like the size of the botnet. Instead, we implemented a bot clone to infiltrate the network and capture all data passing through this system. Furthermore, to measure the size we actively interfered with the botnet to inject the IP addresses of our analysis systems, a method not applied before.

We collected data about the botnet for almost one month between August 6 and September 1, 2009. Our measurement results reveal that the actual size of the botnet is by far bigger than expected, rendering Waledac one of the most efficient spam botnets in the wild. We observed a *minimum* population of 55,000 bots every day, with almost 165,000 active bots on a typical day. In total, we counted more than 390,000 individual bots indicating a similar number of infected machines during the measurement period.

While investigating the botnet, we also witnessed several changes the bot master applied to the botnet to introduce new features like the theft of credentials. We started our research with version 33 of Waledac and finished our observation with version 46. Thus, the botnet is in active development with frequent updates and changes to the core functionality.

Outline. This paper is organized as follows: Section 2 gives a brief background on how the Waledac botnet is structured, the different roles of a bot, and their tasks in the botnet. We present in Section 3 the different experiments we performed while monitoring the botnet and the results we achieved. Finally, we conclude this paper in Section 4.

2 Background

In this section, we briefly describe the setup of the Waledac botnet and its propagation mechanism. More details of the botnet and the technical aspects of its implementation are available in different studies [1, 3, 12, 15, 16].

2.1 Waledac Botnet Structure

The botnet consists of (at least) four different layers, that we describe in the following from bottom to top. Figure 1 shows how these layers are connected and what kind of information is exchanged between them.

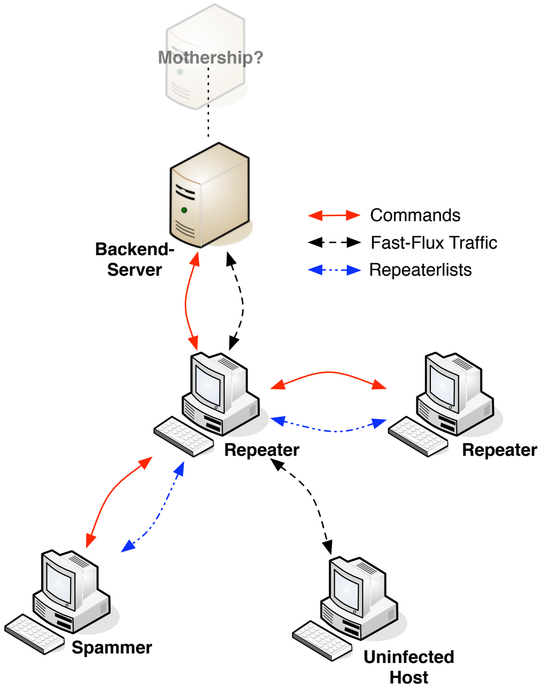

Figure 1: Schematic overview of the Waledac botnet hierarchy.

At the lowest level of the botnet hierarchy are the so called *Spammers*. These systems are used to carry out the spam campaigns. The property that distinguishes Spammers from the other bots in the network, is that they do not have a publicly reachable IP address. That means Spammers are for example located behind NAT routers and therefore cannot be accessed from the Internet directly. The benefit of this property is that although Spammers generate the most attraction due to the massive sending of spam emails, they cannot be easily tracked down.

At the next level are the so called *Repeaters*. These are the entry points for any new bot joining the network, as well as, the place to go for every running bot. For this reason only bots that own a publicly reachable IP address can become a Repeater. A Repeater can be considered as a mediator between the first (lowest) and third (backend) layer of the botnet. Spammers contact the Repeaters to acquire new tasks from the bot master or report the success of previous operations. These requests are relayed to the next layer, the so called *Backend-Servers*. Additionally, the Repeaters act as fast-flux agents for the different Waledac fast-flux domains [9]. That means they also relay HTTP requests of uninfected hosts.

The next level consists of the *Backend-Servers* that answer both the transmitted requests of the Spammers and the fast-flux queries of the Repeaters. As the Backend-Servers are perfectly synchronized and use a webserver software, called *nginx*, that is mainly used for proxy purposes, the assumption of a single server (*mothership*) on top of the botnet is obvious. However, only the analysis of one of the Backend-Servers can prove this assumption right.

Although in related works Waledac is referred to as a pure peer-to-peer botnet, it uses a centralized structure in the upper layers and only the lower ones (Spammers and Repeaters) make up the peer-to-peer part. For this reason, the Repeaters and the Spammers continuously exchange lists of currently active Repeaters. This ensures that any bot at any time has at least one currently running Repeater in his list to join the botnet. As an additional backup, each bot binary contains a hardcoded fail-over URL which itself is hosted within the fast-flux network of Waledac. This URL points to another list of active Repeaters. Thus, if a bot is unable to contact ten Repeaters consecutively on its local list, it downloads a new list from the fail-over URL. Additionally, Repeaters exchange lists of the currently active Backend-Servers. This list is signed with the private key of the botnet herder, to ensure that no attacker can insert his own Backend-Servers into the botnet.

2.2 Propagation Mechanisms

The Waledac bots themselves do not own any built-in propagation mechanisms. That means, infected hosts do not scan their local network for vulnerable systems. Instead, Waledac propagates with the help of social engineering. Hence, Spammers are frequently instructed to send out emails with URLs pointing to current version of Waledac. To increase the probability of infecting new hosts, the self propagation emails are usually masked as greeting cards that host the malicious binary, similar to Storm Worm [10].

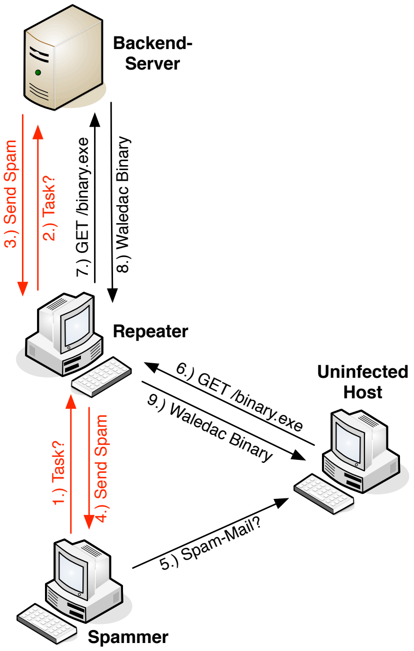

Figure 2: Schematic overview of the Waledac infection cycle.

Figure 2 visualizes the infection cycle of the Waledac botnet. The number at each line indicates the order in which actions are performed. The infection cycle can be summarized as follows: a Spammer frequently queries one of the active Repeaters for new tasks to perform. These queries are relayed to one of the Backend-Servers, that in turn replies with the current task. In this case, the task is to send out spam messages for propagation. An uninfected host that receives one of the emails and follows the embedded link issues a request for the current Waledac binary to one of the fast-flux agents currently assigned to this domain. Again, one of the Backend-Servers transmits the requested content across the agent back to the requesting host. Depending on the reachability of the freshly infected host, it will either show up as a new Repeater or Spammer.

3 Measurements

For our measurements we ran multiple instances of Walowdac on computers at Mannheim university. Considering the single location, all our results for the size should be seen as lower bounds.

3.1 Methodology: Walowdac

Our main objective while investigating Waledac was to find out more about the actual size of the botnet. As the Spammers are not reachable, just crawling the Repeaters does not provide an accurate size of the botnet. To circumvent this problem and to provide a much more accurate number of bots, we implemented a script to imitate a valid Waledac Repeater. The software implements all communication parts of a Repeater, but answers the requests directly instead of forwarding them. We refer to this script as *Walowdac*, as it is a low-interaction Waledac clone.

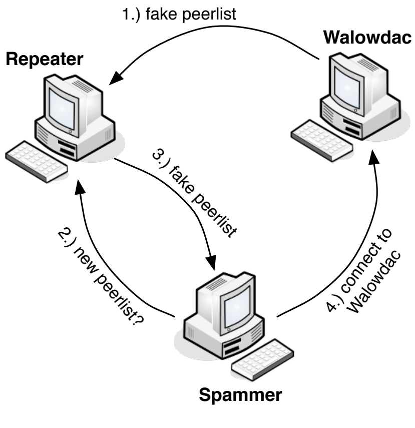

Figure 3: Injecting fake Repeater IP addresses into the botnet.

In order give a more precise lower bound of the Waledac botnet, we push several IP addresses of hosts running Walowdac into the botnet. This is possible as Repeaters do not validate the list of Repeater IP addresses they receive. Thus, anytime our script connects to a Repeater, it sends a list of its own IP addresses to the Repeater. As a result, the IP addresses of our Walowdac systems are propagated throughout the complete botnet and Spammer systems start to connect to us. Figure 3 depicts the single steps performed to distribute the IP addresses of our fake Repeater.

That way, we are able to measure not only the number of Repeaters, but also a large fraction of Spammers. Among the information we store while running our Waledac imitation are timestamps, IP addresses and identification numbers of connecting hosts, Windows and Waledac versions, as well as Spam campaign data distributed through the botnet. With the newer versions of Waledac we also captured stolen credentials of POP3, FTP, and HTTP accounts. During our monitoring period of one month, we collected login data for 128,271 FTP, 93,950 HTTP, and 39,051 POP3 accounts. We have not yet further investigated the stolen credentials, as this is out of the scope of this paper.

All bots within the Waledac botnet can be identified by a *node ID*. This node ID is generated directly after an infection and does not change throughout the lifetime of a bot. Bots embed this node ID in every message they exchange [16] so it is a good candidate to define a uniqueness criterion.

3.2 Results

The results described in this section were gathered between August 6th and September 1st, 2009. For the allocation of IP addresses to countries we used the free version of the GeoIP database maintained by MaxMind [13].

Botnet Size. During the data collection period, we measured 248,983 different node IDs. The maximum number of node IDs on a single day was 102,748 on August 24th. Although the node IDs are randomly generated and should

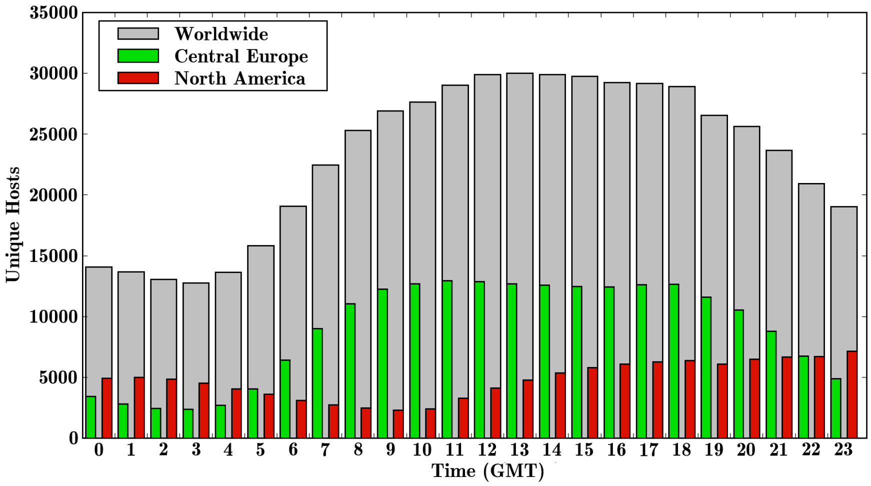

Figure 4: Distribution of running bots according to their location and time on August 24th.

be unique across the botnet, we also monitored several hosts originating in different autonomous systems (AS), using the same node ID. The reason might be collisions in the node ID generation algorithm used by Waledac. With this fact in mind we recalculated the number of bots on August 24th using the node ID and AS as uniqueness criteria, resulting in a total of 164,182 bots. The size of the Waledac botnet we obtained is much higher than previous estimations published by Trend Micro [1] or ESET [2].

Figure 4 shows the hourly number of bots running on August 24th worldwide. For comparison the figure also shows the number of bots located in Central Europe and North America, at the particular hours. The picture also shows the fluctuation (*diurnal pattern*) of running bots due to the different time zones they are located in [5]. Throughout our measurement period, we monitored at least 55,000 node IDs, i.e. active bots, every day.

A cumulative distribution of the bots' IP addresses is shown in Figure 5. We counted all bots monitored during the whole data collection period and again used the node ID and AS as uniqueness criteria. As a result, we counted a total of 403,685 bots. The distribution is highly non-uniform: The majority of bots are located in the IP address ranges between 58.*–99.* and 186.*–222.*. This does not come as a surprise, as most of these IP addresses are managed by the Regional Internet Registries *ARIN* and *RIPE NCC*, which are responsible for the regions North America and Europe, respectively.

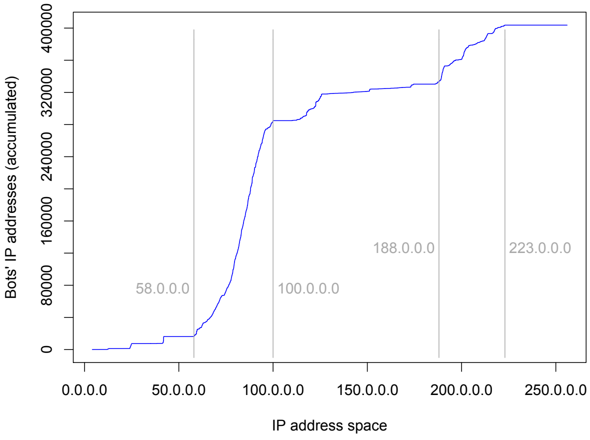

Figure 5: Cumulative distribution of IP addresses infected with Waledac.

This fact is also reflected in Table 6a: Most of the Spammers originated in the US or in Central Europe. The distribution of the Repeaters (see Table 6b) is very similar and differs only in the order of the countries. The main difference is that there are more Repeaters in India than in

Table 6: Top countries in which Waledac bots are located.

Country	# Bots	Percentage
United States	67,805	17.34%
United Kingdom	30,347	7.76%
France	27,542	7.04%
Spain	23,065	5.90%
India	21,503	5.50%
Other	*220,807*	*56.46%*

(a) Spammers

Country	# Bots	Percentage
United States	5,048	19.50%
India	1,456	5.63%
France	1,386	5.36%
United Kingdom	1,348	5.21%
Spain	1,191	4.60%
other	*15,452*	*59.70%*

(b) Repeaters

the United Kingdom and more Spammers in the United Kingdom than in India.

Waledac Versions and Distribution Campaigns. At the beginning of our measurement phase most of the monitored bots were running Waledac version 34. The bot's version number is sent in all its communication packets.

An example of a so called *first* packet is shown in Listing 1. These kind of packets are only sent at the bot's first start after negotiating the session key. The version number is included in the `<v>`-tag [1, 3, 12, 15, 16].

On July 20, 2009, the botnet was ordered to download and run version 36 of Waledac. However, the command was issued just for a couple of hours, thus, systems not online during this time were unable to update. As a result, even two weeks after the update was issued (July 31st), still more than 30 percent of the bots we monitored were running the old version 34. That means, Waledac bots lack a decent update mechanism, since although bots propagate their running version, they are not updated once the command is no longer issued. On July 25th, we monitored the first version 39 bots connecting to our fake Repeater script. The latest version we monitored was 46, which indicates, that the botnet is still actively developed. With version 36, the collection of user credentials was introduced. Table 1 summarizes the distribution of Waledac versions monitored on two different days. At the end of July, most bots are still running version 34 and 36. With the beginning of September this shifted to almost 60% of the bots running the newest version 46.

Listing 1: *First* packet sent after negotiating the session key.

```
<lm>
  <t>first </t>
  <v>34</v>
  <i>4b5da61f8d14e53fe92526694277695e</i>
  <r>0</r>
  <props>
    <p n=”label”>mirabella_site</p>
    <p n=”winver”>5.1.2600</p>
  </props>
</lm>
```

Next to the current version installed on a Waledac bot, bots also send the name of the campaign which distributed the binary. For example, this information is also included in the *first* packets (see Listing 1) – `<p>`-tag with attribute `n="label"`. At the beginning of July the biggest campaigns identified were *birdie6* and *swift*, with 12,5 percent of all infected machines. The current campaigns distributing version 46 are called *spyware*.

OS Version of Infected Machines. Although Waledac bots do not continuously send their operating system version with every packet, but only the first, we managed to capture few of these *first* packets. However, only about 10 percent of all monitored bots established this initial connection to Walowdac. Thus, the results of this measurement provide a coarse overview of the distribution of the operating systems running on infected machines. Table 2 summarizes the operating system codes found in initial pack-

Table 1: Distribution of Waledac version across all monitored bots at the end of July and beginning of September.

Versionscode	31.7.2009 (65,924 Bots)	9.9.2009 (74,280 Bots)
< 33	114 (0.17%)	86 (0.12%)
33	440 (0.67%)	270 (0.36%)
34	**20,718 (31.43%)**	9,344 (12.58%)
35	51 (0.08%)	36 (0.05%)
36	**35,572 (53.96%)**	10,547 (14.20%)
37	2,658 (4.03%)	362 (0.49%)
39	5,681 (8.62%)	1,650 (2.22%)
40	689 (1.05%)	69 (0.09%)
41-45	0 (0.00%)	8,174 (11.00%)
46	0 (0.00%)	**43,742 (58.89%)**

Table 2: Distribution of Windows version codes (June 28th till July 18th.)

code	belongs to	number	percent of bots
5.1.2600	XP (32 Bit)	10,899	90.2%
6.0.6001	Vista (SP1), Server 2008	678	5.6%
6.0.6000	Vista	353	2.9%
6.0.6002	Vista SP2, Server 2008 (SP2)	78	0.6%
5.2.3790	XP (64 Bit), Server 2003	39	0.3%
5.0.2195	2000	27	0.2%

ets. Windows XP still makes up most of all monitored bots, to no surprise.

Spam Campaigns. Throughout the analysis time we monitored different pharmacy and email harvesting campaigns. The harvesting emails advertised cheap watches with the invitation to contact certain emails if interested. Additionally, several Waledac propagation campaigns were observed. For this purpose, the botnet herder used special events, like Valentines or Independence Day, to send out masses of spam messages containing links to Waledac binaries. The same behavior was already observed with Storm.

After each spam run a Spammer reports the status of the transaction for each email. The status can either be *ERR* or *OK*. Thus, it is possible to determine which mail servers did actually accept the incoming email and for which addresses it was rejected. During our monitoring phase we received a total of 662,611,078 notifications, of which 167,784,234 were OK. This gives us an average of 25.32% for the delivery of mails to the recipient's mailserver. In this scope we did not try to determine how many emails actually end up in a user's inbox. A recent experiment by ESET [2] revealed that on average a Spammer using a normal dial-up account sends about 6,500 emails per hour, resulting in about 150,000 spam mails per day.

Taking into account that we monitored at least 10,000 bots online at any time of day, gives Waledac a spam capacity of

$$6,500 * 24 * 10,000 * 0.2532 = 394,992,000$$

delivered mails per day. This number is only a rough estimation and corresponds to the emails actually accepted by the receiving mailservers – theoretically Waledac is able to send more than 1.5 billion spam mails per day. However, this also is only valid for 10,000 bots each hour with our monitoring showing up to 30,000 bots per hour during the daytime. Thus, this number might very well be tripled.

4 Conclusion

In this paper, we showed that it is possible to infiltrate the Waledac botnet by distributing specially crafted peerlists to other Repeaters. As a result, we were able to also collect data from Spammers connecting to our fake system. That way we were able to capture few of the *first* packets send by freshly infected systems. The analysis of these packets revealed that most of the compromised hosts are running Windows XP as an operating system.

We showed that current estimations about the size of the Waledac botnet are far too low. At peaks we measured more than 160,000 bots, whereas ESET [2] for example counted just 20,000 bots. With this in mind, we can estimate that the number of spam emails emitted by Waledac is very high, rendering Waledac one of the most efficient spam botnets currently in the wild. The rapid changes to the malware with new versions showing up almost every two weeks shows that Waledac is still actively developed.

References

[1] Jonell Baltazar, Joey Costoya, and Ryan Flores. Trend Micro: Infiltrating Waledac Botnet's Covert Operations, July 2009.

[2] Sebastián Bortnik. How much spam does waledac send?, 2009. Blog: `http://www.eset.com/`.

[3] Lasse Trolle Borup. Peer-to-peer botnets: A case study on waledac. Master's thesis, Technical University of Denmark, 2009.

[4] Evan Cooke, Farnam Jahanian, and Danny McPherson. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In *Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2005.

[5] David Dagon, Cliff Zou, and Wenke Lee. Modeling Botnet Propagation Using Time Zones. In *13th Network and Distributed System Security Symposium (NDSS)*, 2006.

[6] John R. Douceur. The Sybil Attack. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[7] Felix Freiling, Thorsten Holz, and Georg Wicherski. Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks. In *10th European Symposium On Research In Computer Security (ESORICS)*, 2005.

[8] Julian B. Grizzard, Vikram Sharma, Chris Nunnery, Brent ByungHoon Kang, and David Dagon. Peer-to-Peer Botnets: Overview and Case Study. In *Hot Topics in Understanding Botnets (HotBots)*, 2007.

[9] Thorsten Holz, Christian Gorecki, Konrad Rieck, and Felix Freiling. Measuring and Detecting Fast-Flux Service Networks. In *15th Network & Distributed System Security Symposium (NDSS)*, 2008.

[10] Thorsten Holz, Moritz Steiner, Frederic Dahl, Ernst Biersack, and Felix Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. In *First Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[11] Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, and Stefan Savage. The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff. In *First Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[12] Felix Leder. Speaking waledac, 2009. Blog: `http://www.honeynet.org/node/348`.

[13] Maxmind. Geolocation and Online Fraud Prevention. `http://www.maxmind.com/`.

[14] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *6th Internet Measurement Conference (IMC)*, 2006.

[15] Greg Sinclair. Waledac's communcation protocol, 2009. Blog: `http://bit.ly/MWOA2`.

[16] Gilou Tenebro. W32.Waledac Threat Analysis. Technical report, Symantec, 2009.