The *Dorothy Project*: An Open Botnet Analysis Framework for Automatic Tracking and Activity Visualization

Marco Cremonini
Department of Information Technology
University of Milan
Milano, Italy
marco.cremonini@unimi.it

Marco Riccardi
The Honeynet Project - Italian Chapter
Roma, Italy
marco.riccardi@honeynet.it

***Abstract*—Botnets, networks of compromised machines remotely controlled and instructed to work in a coordinated fashion, have had an epidemic diffusion over the Internet and represent one of today's most insidious threat. In this paper, we present an open framework called *Dorothy* that permits to monitor the activity of a botnet. We propose to characterize a botnet behavior through a set of parameters and a graphical representation. In a case study, we infiltrated and monitored a botnet named *siwa* collecting information about its functional structure, geographical distribution, communication mechanisms, command language and operations.**

I. INTRODUCTION

A *botnet* is a network of compromised machines remotely controlled and instructed to work in a coordinated fashion by one or more management hosts. Botnets are responsible of severe Internet threats, like many Distributed Denial-of-Service (DDoS), spam campaigns, and phishing activity. A large part of today malware epidemics can be related to the spreading activity of botnets [1, 2, 3, 4].

Key to the operations of a botnet is the availability of an efficient communication mechanism between the few hosts in charge of control and management tasks, usually called *Command and Control* (*C&C*), and the many, easily in the order of thousands up to millions in some cases [5], slave hosts usually called *zombies*.

In this paper, we present a new framework called *Dorothy* that aims to automatically perform all the main steps of botnet tracking and to provide real-time data, statistics and graphical representations.

Botnets, since their appearance, have often adopted IRC channels for their communication between C&C hosts and zombies [6]. Subsequently, the focus of botnet analysis shifted on different technologies, in particular to Peer-to-Peer (P2P) botnets [5]. However, IRC botnets are still proliferating and evolving from plain, standard IRC channels to hidden and minimal or encrypted channels. Correspondingly, IRC botnet analysis must evolve to cope with a reduced set of options for estimating a botnet size, to escape from controls put in place by botmasters to avoid infiltration by fake bots, and to manage obfuscation techniques applied for obscuring the semantics of the communication language.

For example, in our analyses, we have found just two C&C hosts over sixteen still relying on plain, standard IRC communications, with no channel restrictions. All the others had disabled information gathering commands and moderated all channels communication. In all cases botmasters were monitoring the joining of new bots to the IRC channel, providing for automatic response mechanisms that permanently banned source IP addresses when anomalies or errors in the command language usage were noted. In some cases, retaliation counteractions in the form of DDoS were activated when the fake bot was spotted.

One method adopted in this work to mitigate the effects of such retaliation actions has been to rely on anonymous proxies and the TOR Network [7] to conceive our IP address during the infiltration process.

II. METRICS FOR BOTNET ANALYSIS

Understanding and characterizing a botnet's structure is still an open issue. Size estimation, hidden structures and the botnet development cycle is hard to investigate, although researchers have recently proposed novel approaches [8, 9].

In general, the first step is to identify the roles of the machines involved. This requires to recognize which hosts are acting as C&Cs, which are *satellites* (i.e., hosts in charge of complementary and supporting operations) and finally which are acting as zombies. Counting zombies which acknowledge to C&C's commands, unfortunately, most of the time, just gives a vague estimation of today's botnet size. The reason is that many of these botnets are rooted by different C&Cs, each one instructing only a branch of the whole network and because often, for a specific activity, just part of the botnet is activated.

In our work, we suggest that a better description could be achieved through a vector of several parameters and a rich graphical representation. In particular, we have identified nine main parameters, as presented in Table 1.

III. THE *DOROTHY* ARCHITECTURE

The architecture of *Dorothy* is composed of several software modules implementing all different phases of the automatic joining to an IRC channel, tracking, analysis and graphical representation of botnet activity (see Figure 1). The ones on which we have mostly concentrated our contribution are: the *Infiltration Module* (IM) and the *Data Visualization Module* (DVM).

TABLE I. BOTNET'S PARAMETERS

Parameter	*Description*
C&C	Number of C&C hosts related to the same botnet
Malwares	Number of malware used for infecting new zombies
C&C Satellites	Number of servers offering complementary features to C&C, such as downloading malwares
IRC Channels	Number of different IRC channels used for C&C-Zombie communications.
Ports	Number of different TCP ports available for IRC communications
Zombies	Number of unique zombies identified as joined to an IRC channel
Hosts	Number of unique host names resolved by zombies through DNS queries. associated to the C&C hosts. This value may give an estimate of the botnet strength with respect to black listing mechanisms.
ALL-Host	Number of unique host names resolved by zombies through DNS queries. This value is likely to be correlated to the amount of spam activity
Mail	Number of different email addresses used as destination by Zombies. As the previous one, also this value is likely to be related to spam activity

A. *Infiltration Module (IM)*

The IM represents the Dorothy's *drone*, a tool that simulates the features of a standard IRC client to permit the joining to an IRC channel. It is implemented with Unix *bash* scripts. Differently from a standard IRC client, a drone must be able to mimic the peculiarities of the specific message protocol and language implemented in botnet communication. For example, usual automatic features embedded in standard IRC clients must be removed, such as automatic join, automatic execution of `LIST` or `WHO` commands, or automatic respose to a `VERSION` request. If executed, such actions, many of the botnets analyzed would have reacted by permanently banning the client IP address. Therefore, the drone should be as stealthy as possible and should not reveal its own information. A requirement for the Dorothy's drone was to establish connection with C&C hosts trough an anonymization service to protect our identity from retaliation actions. The drawback of this solution is that it increases the transmission delay, which could appear to a C&C as an insufficient quality of transmission and provoke the disconnection of the client.

B. *Data Visualization Module (DVM)*

Studing different means of visualizing information related to botnets behavior is a key goal of the project. The DVM automatically provide all graphs by means of *AfterGlow* scripts [10]. Data available for visualization are received by the DVM from previous modules and are characterized on a three dimensions matrix whose parameters are: *Source*, *Service*, and *Target*. Given this representation, the DVM first produce a *link graph* of the relationships between zombies and other hosts. Nodes properties in the link graph are identified by color, shape, and dimension. Some data aggregation are performed, such as for IP addresses, data gathered from IRC communication, and mail traffic.

It is also possible to merge several dumps of network traffic and to process them as a whole using the same visualization information process used to produce the link graph for a single zombie.

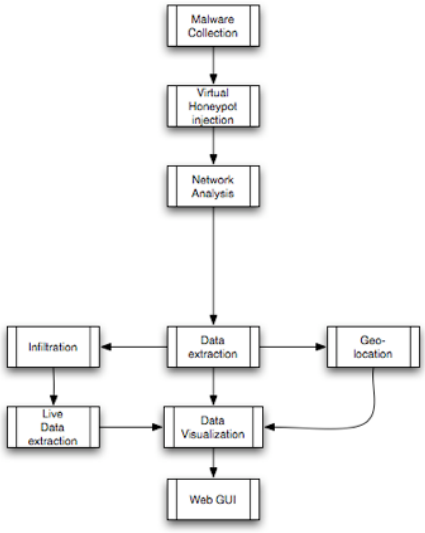

Figure 1. The modular architecture of Dorothy.

IV. RESULTS

The Dorothy framework was tested for a period of 27 days between January and March 2009. This was the first test of the system with live malwares. During this period, Dorothy downloaded 3900 (304 unique) malware binaries (562,657 Mb) and our honeypot was successfully compromised 5291 time. All the malware binaries were downloaded from 2210 unique IP address, using 2275 different source TCP ports. In particular, a large proportion of malwares were downloaded by few sources (i.e., three IP addresses accounted for about 80% of malwares); 16 different C&Cs were identified and 50 IRC channels have been successfully infiltrated by the drone.

Of the monitored C&Cs, four plainly responded to `WHO` and `LUSER` IRC commands, showing all zombie identities. Four instead simply ignored any command. The others, in some cases accepted a command, but eventually disconnected the drone and permanently banned the IP address. Interestingly, in four cases it was possible to intercept the short timeframe in which the botmaster disabled the channel moderation, which let us watching and recording all the conversations between C&Cs and zombies. These short intervals were probably a sign of a reconfiguration activity performed by the botmaster. About the command issued, nearly half of the cases the connection was encrypted, and for those unencrypted channels, the great deal of activity was devoted to enlarge the botnet. From the acquired data, it was possible to recognize three different Spam Centers and a total of 3157 unique email addresses.

The infiltration process has recorded 8992 unique public network addresses. These hosts represents zombie connected

to the monitored C&Cs. Of these, 44,5% were found to be unresolved host name.

V. CASE STUDY: THE *SIWA* BOTNET

The *siwa* botnet has been monitored in detail during the test. It is formed by five C&C hosts located in China (2), Canada (2) and Holland (1). The botnet makes use of seven IRC channels called: `#siwa`, `##russia##`, `##loose`, `##pi##`, `#bb`, `#ns`, and `#q52`. The first two were encrypted, but from the others it was possible to monitor the ongoing activity. For example, one command issued by the C&C was:

```
##pi## :* ipscan s.s dcom2 -s ][ * wormride on -s
][ * download http://72.xxx.xxx.xxx/mb2.exe -e –s
```

This was meant to instruct the zombies to start a spreading activity towards the hosts in their own network (`s.s` is a shortcut for 255.255.0.0) using the `dcom2` module. Furthermore, it enabled the exploit module `wormride` (*wormride* is a known Internet worm used to compromise Windows DCOM services) and then downloaded and executed the file `mb2.exe` from the specified HTTP site.
As a consequence, the zombies sent the acknowledgment:

```
72.xxx.xxx.xxx:2293 -->
:QfNUXNcm!~xqbmgz@92.xxx.xxx.xxx PRIVMSG
##RUSSIA## :-041- Running FTP wormride thread
```

```
72.xxx.xxx.xxx:2293  --> :Tdkzdtwh!~bxoluj@mna75-
4-82-225-77-1.yyy.yyy.net PRIVMSG ##russia## :-
04wormride- 1. tftp transfer to 82.xxx.xxx.xxx
complete.
```

These two messages mean that zombies started their spreading activity using the new downloaded exploit module after its thread activation, uploading it via the *tftp* protocol. From this log fragment, it is also possible to see the different usage of IRC channels.

Next we investigated which satellite hosts were supporting the five C&Cs. We identified 37 different satellites used to distribute different malwares through HTTP connections. Further analyses showed that the C&C hosts adopted different configurations. For example, one of the Chinese C&C acted as both an IRC commander and as a web satellite. It configured TCP ports 2293, 2569, 2938, 3240, and 3838 for the IRC communications and TCP port 80 to share malwares with its own zombie community. Differently, the other Chinese C&C configured just TCP port 65520 for IRC communications, but acted also as a Spam Center, rather then a satellite (i.e. about 99% of DNS queries and email deliveries were generated by this host). The Holland C&C, instead, tried to disguise IRC communication by configuring the channel to respond on TCP port 80.

Associated to the *siwa* botnet, we discovered 42 unique host names. Trying to resolve these host names failed in just four cases. The remaining were all correctly resolved to IP addresses referred to either C&C or satellite hosts.

For those IRC channels that did not make use of encryption or obfuscation techniques, it was possible to intercept conversations between zombies and C&C hosts. This way, we identified 4346 unique IP addresses that were acting as zombies. Interestingly, by monitoring each time a C&C instructing its zombies to download a new malware, we discovered that on average, this happens every 6 hours.

Figure 2 shows the graphical representation of locations and links between C&C hosts and their satellites on a Google Map. Selecting a location on the map, real-time statistics of the corresponding C&C or satellite host are visualized.

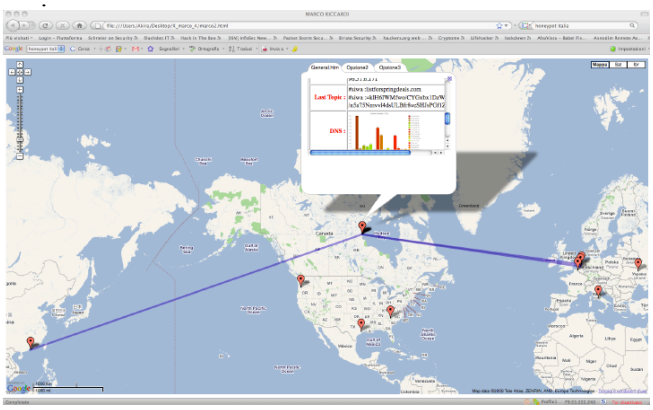

Figure 2. Active representation of *siwa*'s C&C hosts and their satellites.

REFERENCES

[1] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. In Proceedings of the Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI'05), USENIX, Cambridge,MA, July 2005. pp. 39–44.

[2] M.A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In Proceedings of the 6th ACM SIGCOMM Internet Measurement Conference, Rio de Janeiro, Brazil, October 2006.

[3] Honeynet Project. Know your Enemy: Tracking Botnets, March 2005. http://www.honeynet.org/papers/bots.

[4] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir. A Survey of Botnet Technology and Defenses In Proceedings of Cybersecurity Applications & Technology Conference For Homeland Security (CATCH), Washington, DC, March 2009. pp. 299-304.

[5] 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 Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, USENIX, Berkeley, CA, 2008. pp. 1–9.

[6] J. Zhuge, T. Holz, X. Han, J. Guo, and W. Zou. Characterizing the IRC-based Botnet Phenomenon. Department for Mathematics and Computer Science, University of Mannheim, TR-2007-010 ResearchPaper, 2007.

[7] R. Dingledine, N. Mathewson, and P. Syverson, Tor: The second-generation onion router. In Proceedings of the 13th Usenix Security Symposium, USENIX, San Diego, CA, August 2004, pp. 303-320.

[8] 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 Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots'07), USENIX, Berkeley, CA, April 2007.

[9] J. Leonard, S. Xu, and R. Sandhu. A First Step towards Characterizing Stealthy Botnets. In Proceedings of the International Conference on Availability, Reliability and Security, Fukuoka, Japan, March 2009. pp.106-113.

[10] AfterGlow. http://afterglow.sourceforge.net/