The Journal of Systems and Software 86 (2013) 707–715

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

BotMosaic: Collaborative network watermark for the detection of IRC-based botnets

Amir Houmansadr[a,*], Nikita Borisov[b,*]

[a] Computer Sciences Department, The University of Texas at Austin, Austin, TX 78758, USA
[b] Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Urbana, IL 61820, USA

ARTICLE INFO

Article history:
Received 7 March 2012
Received in revised form 12 October 2012
Accepted 2 November 2012
Available online 9 November 2012

Keywords:
Botnet detection
Botmaster traceback
Traffic analysis
Network flow watermarking
Network security

ABSTRACT

Recent research has made great strides in the field of detecting botnets. However, botnets of all kinds continue to plague the Internet, as many ISPs and organizations do not deploy these techniques. We aim to mitigate this state by creating a very low-cost method of detecting infected bot host. Our approach is to leverage the botnet detection work carried out by some organizations to easily locate collaborating bots elsewhere.

We created BotMosaic as a countermeasure to IRC-based botnets. BotMosaic relies on captured bot instances controlled by a watermarker, who inserts a particular pattern into their network traffic. This pattern can then be detected at a very low cost by client organizations and the watermark can be tuned to provide acceptable false-positive rates. A novel feature of the watermark is that it is inserted collaboratively into the flows of multiple captured bots at once, in order to ensure the signal is strong enough to be detected. BotMosaic can also be used to detect stepping stones and to help trace back to the botmaster. It is content agnostic and can operate on encrypted traffic. We evaluate BotMosaic using simulations and a testbed deployment.

1. Introduction

A *botnet* is a network of compromised machines, *bots*, that is controlled by one or more *botmasters* to perform coordinated malicious activity. Botnets are among the most serious threats in cyberspace due to their large size (Ramachandran and Feamster, 2006). This enables the bots to carry out various attacks, such as distributed denial of service, spam, and identity theft, on a massive scale.

Botnets are controlled by means of a command-and-control (C&C) channel. A common approach is to use an Internet Relay Chat (IRC) channel for C&C: all the bots and a botmaster join a channel and the botmaster uses the channel to broadcast commands, with responses being sent back via broadcast or private messages to the botmaster. The IRC protocol is designed to support large groups of users and a network of servers to provide scalability and resilience to failures, thus it forms a good fit for providing a C&C infrastructure. Because of their simple design and deployment, IRC botnets have been widely used by cybercriminals since 2001 (Kharouni, 2009). Some botnets use a more advanced structure, with bots communicating directly with each other in a peer-to-peer fashion, but recent studies show that *many existing botnets use the IRC model* because of its simple-yet-effective structure (Kharouni, 2009; Zhuge et al., 2007). In this research we focus on the IRC botnets.

Much research has been devoted to the detection of IRC botnets (Binkley and Singh, 2006; Ramachandran et al., 2006; Karasaridis et al., 2007; Collins et al., 2007; Villamarín-Salomón and Brustoloni, 2009; Zilong et al., 2010). However, most effective detection techniques are complex and have potential to generate false positives. This means that organizations with a large security budget are able to find potential bot infections and disable, investigate, and disinfect affected machines. Organizations with less developed IT practices, as well as home users, however, remain vulnerable to bot infections and provide a fertile ground for botnets, allowing them to remain strong.

We propose a technique that follows a service model. It leverages the efforts of one organization to capture and instantiate bot instances to provide low-cost detection of bots in other networks. We develop BotMosaic – a watermark that, when inserted into the communication between the captured bots and an IRC server, creates a pattern that is observable at other sites hosting botnets. The pattern can be recognized simply by observing the timings of the packets in a given flow, thus the detection can be carried out at a large scale by border routers. By inserting an artificial pattern, we can ensure that false-positive rates are very low, enabling automated actions to disconnect infected bots. Since only packet

* Corresponding authors. Tel.: +1 217 722 1761.
E-mail addresses: amir@cs.utexas.edu (A. Houmansadr), nikita@illinois.edu (N. Borisov).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.11.005

timings are used, BotMosaic works even when the botnet uses encrypted connections to the IRC server.

The watermark will be visible on all connections between the bots and the IRC server. It will likewise appear in the connection from the botmaster to the IRC server. Botmasters typically use stepping stones (Zhang and Paxson, 2000) to hide their true location. The watermark can be used to detect such stepping stones and aid in botmaster traceback.

A novel and unique feature of our watermark is that it is *collaborative*: the watermark is inserted simultaneously into the flows of all captured bots. This is in contrast to past watermarks that affect a single flow at a time (Wang and Reeves, 2003; Wang et al., 2005, 2007; Pyun et al., 2007; Yu et al., 2007; Houmansadr et al., 2009b; Ramsbrock et al., 2008). The collaborative feature amplifies the effect of the watermark and is necessary to create a timing pattern that is recognizable among the noise generated by traffic from other bots. In other words, this collaborative behavior allows a BotMosaic watermark to persist on the watermarked flows even after they are mixed with other flows in the botnet's C&C channel.

In summary, BotMosaic has the following unique features as compared to previous approaches: (1) BotMosaic is implemented by one organization, and can be used as a *low-cost* service by other organizations, i.e., *clients*. A client organization only needs to deploy the low-cost watermark detectors of BotMosaic on their border routers. This is in contrast to other approaches that suggest each organization to deploy its own, resource-intensive botnet detection mechanism. (2) A client organization can use BotMosaic to detect various instances of bots simultaneously, without the need to modify its BotMosaic detectors for different botnets. The BotMosaic watermarkers use different watermark signals for different instances of botnets. (3) Each client organization can detect not only the bot infected machines, but also the botmasters and stepping stones hosts residing *inside* their networks.

We analyze our scheme using simulations and experiments on PlanetLab (Bavier et al., 2004). We find that we can achieve a high rate of detection with few false positives using a watermark applied to captured/imitated bots that comprise a small fraction of the botnet, with a detection time of about a minute.

The rest of the paper is organized as follows: Section 2 describes previous work on IRC botnet detection and reviews past work on network flow watermarking. Section 3 describes the overall detection framework used by BotMosaic. Section 4 describes the detailed structure of the BotMosaic collaborative watermark. Simulations and implementation results are presented in Section 5. Section 6 offers a brief discussion of some additional issues, and Section 7 concludes the paper.

2. Related work and motivation

The primary goal of the BotMosaic is to detect bot-infected machines inside a network of interest, e.g., an ISP. The literature on this can be divided into *host-based* and *network-based* approaches. Host-based approaches analyze the information on hosts of the network; this is not easy to deploy on all hosts, especially in organizations where computers are not centrally managed. BotMosaic falls in the network-based category.

Network-based detection mechanisms aim to detect bot infected machines by analyzing the network traffic information. These mechanisms mainly are classified into two categories: *traffic signature* schemes and *traffic classification* schemes. The traffic signature approaches use the captured bots to develop signatures for each botnet instance; they have widely been used for IRC botnet detection (Binkley and Singh, 2006; Karasaridis et al., 2007; Goebel and Holz, 2007). As an example, Binkley and Singh (2006) combine IRC statistics and some TCP metrics to generate signatures that can be used to detect the infected machines. Traffic classification approaches are based on gathering network traces and clustering them in order to detect botnets based on their behavioral difference with the normal traffic (Ramachandran et al., 2006; Villamarín-Salomón and Brustoloni, 2009; Collins et al., 2007). As an example, Villamar et al. use Bayesian methods to isolate centralized botnets, based on the similarity of their DNS traffic with those of some known DNS botnet traces (Villamarín-Salomón and Brustoloni, 2009).

In this paper we consider a third approach for performing network-based bot detection. BotMosaic uses *network flow watermarking* to mark the botnet traffic, resulting in low-cost mechanisms for the detection of bots and botmasters. Network flow watermarking is a technique that actively perturbs the traffic patterns of a network flow to insert a watermark inside them that can later be detected. Flow watermarking has been used to detect stepping stones, as well as to compromise anonymous communication (Wang and Reeves, 2003; Wang et al., 2005, 2007; Pyun et al., 2007; Yu et al., 2007; Houmansadr et al., 2009b). Existing techniques, however, cannot be applied to the problem of bot/botmaster traceback for two reasons. First, they are designed to work on long-lived flows; typically, hundreds of packets are necessary to detect the presence of a watermark. Botnet communication, however, tends to be short-lived, with only a few packets sent from each bot. Furthermore, a watermark that is applied to a single bot-to-botmaster/botnet communication will be overwhelmed by traffic from other bots that will be aggregated along the same stepping stone connection. Although some of the existing watermarks are designed to resist a reasonable level of chaff, they do so by increasing the length of the watermark and thus cannot be used for botnet traceback in practice.

More recently, Ramsbrock et al. (2008) designed a watermark specifically targeted to the task of botmaster traceback. Their watermark works by adding extra whitespace at the end of IRC messages sent by the bots. They also adjusted the timings of packets in order to improve detection ability. Though an important first step, the whitespace watermarking approach has several serious limitations. Whitespace watermarking only works well in the presence of low rates of chaff – less than 0.5 packets/s – whereas even in a small-size botnet, an aggregate response from all the bots would create a significantly higher chaff rate. Whitespace watermarking is also fragile to repacketization or retransmission of packets, as such events can cause it to lose timing synchronization. Finally, whitespace watermarking relies on modifying the *contents* of the messages sent by the bots, which can be difficult if encrypted connections are used.

Network flow watermarking. Recently, researches have proposed to use network flow watermarks in different applications. Wang et al. were the first to borrow the watermarking idea from multimedia literature to do active traffic analysis (Wang and Reeves, 2003). They use QIM watermarks over inter-packet delays (IPD) of the network flows, providing a more efficient scheme for the detection of stepping stone attacks compared to similar passive detection schemes (Wang et al., 2002).

To make the detection scheme robust to packet-level modification several watermarking schemes suggest an interval-based approach (Pyun et al., 2007; Yu et al., 2007; Wang et al., 2007). In particular, Pyun et al. (2007) proposes an interval-based watermark for detection of stepping stones which is robust to repacketization. A similar interval-based scheme is proposed in Yu et al. (2007) that utilizes packet rates for watermarking. Wang et al. propose another interval-based flow watermark to compromise anonymity in low-latency anonymous networks (Wang et al., 2007). Kiyavash et al. (2008) introduce a multi-flow attack that is able to compromise the interval-based watermarks of Pyun et al. (2007), Wang et al. (2007) and Pyun et al. (2007). Houmansadr et al. (2009b) use a