Your Botnet is My Botnet: Analysis of a Botnet Takeover

Stone-Gross, Cova, Cavallaro, Gilbert, Szydlowski, Kemmerer, Kruegel & Vigna*

February 9, 2010

Introduction

Malware is a major security problem on the Internet. The most dangerous forms of malware turn the victim’s computer into a bot that can be controlled at the attacker’s will, and add the victim’s computer to a botnet of similarly infected computers, all of which can be controlled simultaneously by the attacker (or botmaster) via a command & control (C&C) server. Previous research indicates that botnets are used for criminal activity, such as large-scale email spamming, launching denial-of-service (DoS) attacks against other computers on the Internet, and stealing personal information from the owners of infected computers with the intent of committing identity theft.

A large research effort is underway to analyse the structure and activity of botnets. Previously, researchers have used passive analysis techniques to monitor botnets; these techniques observe the secondary effects of botnet activity. Some methods that have been used include counting spam emails or DNS lookups that most likely originated from infected computers, and monitoring ISP network traffic for suspicious activity. These techniques, however, are unreliable; counting spam emails and DNS lookups is an inaccurate way of sizing a botnet since it is possible for just part of the botnet to be active at any given time, and similarly monitoring ISP traffic can result in legitimate traffic being identified as suspicious.

The authors of this paper instead chose to actively analyse the Torpig botnet: by reverse-engineering parts of the bot code, they were able to infiltrate the botnet and seize control of the C&C infrastructure from the real botmaster. They logged all traffic between bots and the C&C servers and analysed the data transmitted between them.

Background

The Torpig bot code is shipped with a rootkit named Mebroot. Mebroot is particularly dangerous because it installs itself in the master boot record (MBR) of the victim’s hard drive, making it very difficult for anti-virus software to detect. A computer becomes infected with Torpig in the following way:

1. The victim visits a web page hosted on a vulnerable web server. The page contains an <iframe> linking to a Mebroot download server...
2. ...that is loaded by the victim’s web browser. Mebroot attempts a number of exploits against the browser; if any are successful, Mebroot is silently downloaded and installed.
3. Mebroot installs itself in the MBR and uses other mechanisms to evade detection by anti-virus software. It then downloads Torpig DLLs from a Mebroot C&C server, and injects them into specific processes that handle passwords, bank account details and other sensitive data.
4. Torpig periodically reports back any lucrative data to a Torpig C&C server.

Taking Control

A critical aspect of any botnet is how its bots locate their C&C server. Torpig uses the concept of domain flux: from time to time, bots independently use an algorithm to deterministically generate a list of domain names; Torpig then tries to contact a C&C server at each of the domain names, working its way through the list until it finds a genuine C&C server. The authors of this paper reverse-engineered

*Summarised by Chris Novakovic
Torpig’s domain generation algorithm and registered three weeks’ worth of domain names that would be used in future by the botnet; the Torpig botmasters had forgotten to register these domain names, or were simply too lazy to do so.

Next, the authors set up a web server to act as a *sinkhole* to collect data from bots. They reverse-engineered the C&C communication protocol and learned how to correctly identify their sinkhole as a C&C server (to ensure that bots would communicate with their server and not query the next domain name in the list), and how to decrypt data sent by bots (which was encrypted using a very weak symmetric-key algorithm, and was trivial to break as the key was sent in-the-clear).

**Botnet Analysis**

One of the authors’ goals was to accurately estimate the size of the Torpig botnet; other researchers’ attempts at measuring the size of other P2P botnets (such as Storm) were inaccurate. The authors identified that each infected computer controlled by Torpig is assigned a unique ID that is sent back to the C&C server along with every piece of data. By counting the number of unique IDs observed, the authors reliably estimate that there are around 180,000 bots in the Torpig botnet; between 100,000 and 125,000 of these are active at any given time, and up to 1,800 new infected computers join the botnet every day.

The authors also identified that Torpig infections were grouped according to a build ID in the header of each piece of data sent to the C&C server. The authors speculate that this build ID is being used to split the botnet into different sections, as their analysis revealed that it had seemingly no other purpose. Dividing up the botnet in this way allows the botmasters to “rent out” sections of the botnet to other criminal organisations, effectively handling day-to-day administration of the botnet for other organisations that cannot or do not want to maintain their own botnet infrastructure.

**Data Analysis**

Just over 15 million data items were sent to the sinkhole over a period of ten days. The vast majority of these (almost 12 million) consisted of HTML `<form>` data, containing usernames and passwords for authenticating to services such as web-based email and online banking. The contents of the sinkhole revealed that the bots had reported around 8,000 logins for hundreds of financial institutions such as Paypal, Capital One and E*Trade. Credit card details were also skimmed from victims’ web browsers using a simple algorithm; this yielded 1,660 credit card numbers. Authentication details for financial services are highly sought after in the criminal world, with credit card numbers being sold for up to $25 and bank accounts up to $1,000 — if all of this data were sold for its maximum potential value, the botmasters would have turned a profit of $8.3 million in ten days.

Torpig opens a number of ports on the victim’s computer that can be used as proxies for sending spam emails. Many anti-spam techniques rely on publicly-available blacklists of IP addresses known to be distributing spam; as the majority of the bots were running on residential Internet connections, their IP addresses were unlikely to be present in these blacklists and the spam emails they were sending were more likely to be successfully delivered to their targets.

50,000 of the bots were running across residential cable or DSL Internet connections, and 17,000 across corporate connections. The authors estimate that the total bandwidth available to the bots on residential connections was 17Gbps, with the bandwidth available to bots on corporate networks likely to be significantly higher. This would be more than enough to launch a major DoS attack against other host on the Internet, most likely for the purpose of extortion.

**Conclusions**

From their active analysis of Torpig, the authors conclude that botnets exist because many computer users have poor or non-existent security procedures, making it possible for their computer to be easily turned into a bot. They suggest that this is a cultural problem, and that the size of botnets could be cut considerably if computer users were better educated about online security. The authors also identify that is difficult to communicate with domain name registrars, hosting companies and law enforcement agencies — as well as individual users, it is also the responsibility of these organisations to be proactive about limiting the existence of botnets by responding to abuse reports quickly and efficiently.