Lecture 29: Bots, Botnets, and the DDoS Attacks

Lecture Notes on "Computer and Network Security"

by Avi Kak (kak@purdue.edu)

April 12, 2017
4:25pm

Goals:

- Bots and bot masters
- Command and communication needs of a botnet
- The IRC protocol and a command-line IRC client
- Freenode IRC network for open-source projects and the WeeChat IRC client
- **Python and Perl code for a command-line IRC client**
- **Python and Perl code for a mini-bot that spews out spam**
- DDoS attacks and strategies for mitigating against them
- **Using IoT devices to launch crippling DDoS attacks**

CONTENTS

29.1: BOTS AND BOT MASTERS

- Earlier in Lecture 22, we focused on viruses and worms. Typically, viruses and worms are equipped with a certain fixed behavior. Any time they migrate to a new host, they try to engage in that same behavior.

- A bot, on the other hand, is usually equipped with a larger repertoire of behaviors. Additionally, and perhaps even more importantly, a bot maintains, directly or indirectly, a communication link with a human handler, known typically as a bot-master or a bot-herder.

- The specific exploits that a bot engages in at any given time on any specific host depend, in general, on what commands it receives from some human. **You could say that a basic characteristic of a bot is that it does the bidding of the bot master.**

- A bot master can harness the power of several bots working together to bring about a result that could be more damaging than

what can be accomplished by a single bot (or a worm or a virus) working all by itself. The bots working together could, for example, mount a **distributed denial of service (DDoS)** attack that would be much more difficult to protect against than a regular denial of service attack (DoS) we talked about in Lecture 16. Several bots working together would also be more effective in spreading virus and worm infections, and in corrupting the machines with spyware, adware, etc. Additionally, it would be much more difficult to squelch spam if it is spewing out simultaneously from several bots at random locations in a network. [A botnet may infect millions of computers. The botnet dismantled most recently, Rustock, was believed to have infected close to a million computers. This botnet as a whole was sending several billion mostly fake-prescription-drugs related spam messages every day. Rustock was dismantled by Microsoft through a court-ordered action that shut down the botnet's command and control servers that Microsoft was able to locate in several cities in the United States. While the dismantling of Rustock is indeed a major triumph, its human handles have not yet been identified (to the best of what I know).]

- Being generally a more powerful piece of software, a bot may also exhibit greater ability to adapt its behavior to its environment. As a case in point, a bot may prove more adept at understanding the security features of a host and at weakening them for its own benefit. To illustrate, some folks think of the Conficker worm (see Lecture 22) as a bot because of its advanced communication abilities and, even more particularly, because of its ability to prevent a host from contacting security agencies for the purpose of downloading updates that may prevent the worm from operating.

- A collection of bots working together for the same bot-master constitutes a **botnet**.

- At Purdue University, we have recently developed a new approach to the detection and isolation of botnets in a computer network. Our method is based on a probabilistic analysis of the temporal co-occurrences of malicious activities in the different computers in a LAN. On the basis of the results obtained on simulated botnet data and *on actual network traces*, we believe this approach is more powerful than the other approaches that have been developed to date. Our approach is described in the paper cited on the next page.

- What makes our approach particularly powerful is that it does not make any assumptions about the mode of command and control used in the botnets. Most of the competing approaches are based on specific assumptions regarding how the bots in a botnet communicate with one another and with the botmaster.