MobileBot, GameBot, ?Bot: The Security Threats To and From the Intelligent Electronics Devices

Gang Xu, Jay Jayawardena, Cristina Serban, Gustavo de los Reyes, Gokul Singaraju, Krishna Sistla and Phi Nga Hoang

AT&T Chief Security Office, 200 Laurel Ave, Middletown, NJ 07748, USA
{*gangxu, tj, cserban, gdelosreyes, gs244f, ks4308*}*@att.com*

1 Introduction

Botnets have become one of the most prominent threats to internet security. To create botnets, hackers infect millions of computers, or so-called bots, and orchestrate them to launch a variety of attacks such as identity theft, spamming, and distributed denial-of-service. In spite of the tremendous efforts from the internet security community to suppress botnets, such as taking down the hosting facilities [2], the problem continues to increase. As disclosed in McAfee threats report [7], almost 12 million new IP addresses all over the world were taken control of by botnet controllers in the first quarter of year 2009, a 50-percent rise over last year.

To date, botnets have been virtually only a PC problem. However, we envision that this will no longer be true in near future. What drives the potential change is the exponentially increasing popularity of intelligent electronic devices such as smartphones/pocketPCs, gaming consoles, networked televisions and even digital photo frames. Many of these devices have considerable computing power. For example, the 3GS iPhone is equipped with ARM11 CPU that runs at 412MHz in power saving mode and 128 DDR memory. Gaming consoles are even more powerful than many PCs. Furthermore, all of these devices can be connected to the Internet via WiFi and/or the cellular network. In addition, these devices run a full-fledged operating system, e.g., Windows Mobile, Symbian or Linux, and are highly programmable. That said, these devices are in fact powerful computers comparable to PCs. Despite this fact, most people still perceive them as dumb consumer electronics and may not pay sufficient attention to their security.

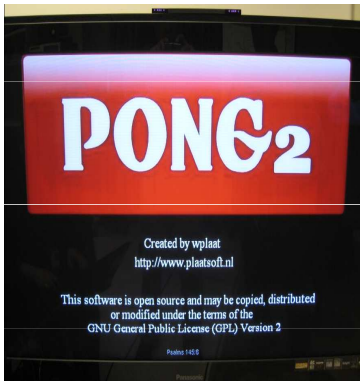
The combination of high computing resources, network connectivity, and ubiquity makes these intelligent devices perfect targets for botnets. In this paper, we demonstrate the feasibility of creating botnets using these intelligent devices and exploiting them to launch attacks. Our experimental botnet is composed of one PC bot, one smartphone bot and one gaming bot. But theoretically all intelligent networked devices such as digital picture frames or even home appliances like refrigerators, etc. can be part of such a botnet. The bots are connected through the Internet Relay Chat (IRC) protocol and are commanded via a smartphone-based controller to execute distributed denial-of-service (DDoS) attacks. The experiment shows that such hybrid botnets can cause a devastating effect on networks and services. To defend against such an attack, we show a detection method based on network flow analysis.

The rest of the paper is organized as follows: Section 2 describes the experiment of creating the hybrid botnet and using it to run attacks. Section 3 discusses the method to detect the botnet attack. We conclude the paper in Section 4.

2 Hybrid BotNet

As a proof of concept, we created a hybrid botnet consisting of three bots running on a PC bot (PCBot), a smartphone (MobileBot) and a gaming console (GameBot), respectively. These bots are connected through the IRC protocol to launch a distributed denial-of-service attack.

MobileBot



GameBot

Figure 1. MobileBot (left) and GameBot (right).

2.1 Building the Bots

A bot is essentially a combination of two components: an IRC client used to communicate with the controller/attacker, and a malicious agent that executes commands received from the attacker. Due to the difference in the operating systems, we implemented the three bots separately.

- **PCBot**
  Our PCBot runs in a Windows XP (32 bit) virtual machine with 1G DRAM. The host system is a PC workstation with an AMD 64 bit dual-core processor and a Gigabit ethernet card, also running 32 bit Windows XP. We developed the PCBot by modifying a popular bot program, RBot [4]. Instead of using the built-in attacks, we implemented a customized UDP flood attacking module.

- **MobileBot**
  We implemented the MobileBot using a state-of-the-art smartphone running Windows Mobile 6.1 Professional Edition. The phone is equipped with a Qualcomm MSM 7201 400 MHz CPU, 128 MB RAM and 256 MB flash memory. The program is written in Microsoft C# and developed using Microsoft Visual Studio 2008. In order to entice users to install the MobileBot, we camouflaged the Mobilebot program with a Music download frontend shown in Figure 2.

  The IRC client in the MobileBot and the GameBot discussed next is essentially ported from a free java based IRC client [5]. As for the attacking agent, UDP attacks are preferred over TCP for the sake of high volume and difficulty of defense. However, since the gaming console operating system does not support a UDP stack, we had to implement a TCP-based flood, e.g., syn flood. To enhance the volume of traffic, the attacking agent is multi-threaded and the attacker can control the number of threads to be used for the attack.

- **GameBot**
  The GameBot is implemented using state-of-the-art gaming console with an IBM PowerPC 729 MHz chip (Broadway) and 128M RAM. Since the game developer kit is not open to the public, we used a free DevKitPro [1] to develop the bot on a Windows XP PC and cross-compiled it for the game console. The program is written in C.

  As with the MobileBot, the GameBot consists of a thin IRC client and a multi-threaded attacking agent that runs TCP-based flood. What differentiates GameBot from the MobileBot and the PCBot is that the gaming console is a closed system. By default, the gaming console is locked such that only authorized games (e.g. those purchased with licenses) can be run. This makes it difficult to mount a bot program on the gaming system. We solved this problem by exploiting a known buffer overflow vulnerability in one of the commercially available games. By making a special movement at a particular scene of the program, the vulnerability is triggered to render the system control. Then, we are able to mount and execute the bot program. Similar to the MobileBot, we need to hide the bot. This was done by embedding the bot in an open-source game such that when the game is loaded and played, the bot

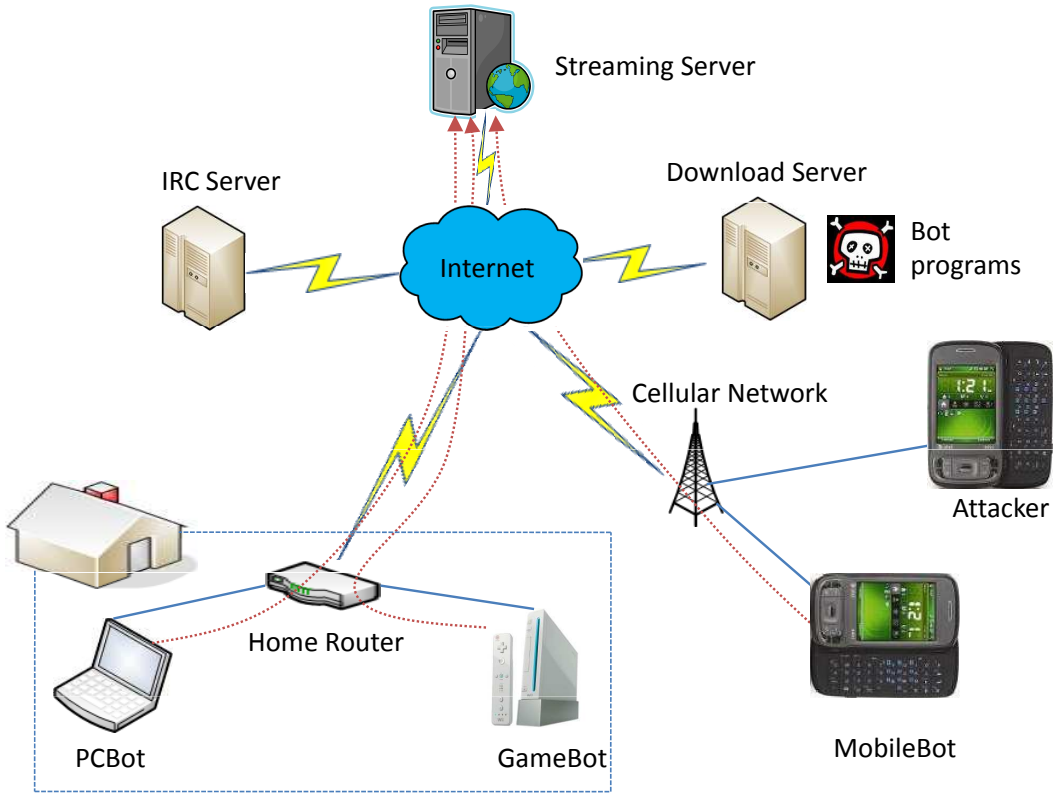


Figure 2. Launching Hybrid Botnet Attack. The hybrid botnet is composed of a PCBot, a MobileBot, and a GameBot. The attacker controls them via the IRC protocol to launch a DDoS attack to a video stream server.

runs covertly in the background. Our prototype used a simple free game, called Pong2 [3] to wrap the GameBot. Figure 2 shows the screen capture of the Bot running in the background of the Pong2 game.

2.2 Creating the Botnet

Once the attacker develops the bot programs, the next step is to spread them and infect as many systems as possible to create a large hybrid botnet. There are many ways to spread malicious code. A common method is to entice victims to download the code by claiming it to be an attractive and legitimate program.

The PCBot is the easiest to distribute and can spread in the same way as viruses. A more efficient way to distribute the MobileBot is through a short message (SMS) that includes the download link for the bot. Distribution of GameBot is similar except that the gaming system must have been unlocked as we discussed before. We take advantage of users that are highly motivated to unlock their gaming systems in order to play free games.

Before distributing the bots, the attacker needs to find or establish an IRC server and register a chat channel as the master. After the bots join the network, the attacker can control them to launch attacks.

2.3 Launching the Attack

We tested our hybrid botnet by using it to launch a DDoS (also known as a flood) attack to a streaming server running VLC media player [6]. All the bots are connected to an UnrealIRCd IRC server and controlled from a jmIRC client running on a smartphone. Both the PCBot and the GameBot are connected via Ethernet LAN while the MobileBot and the attacker's botnet controller are connected via a cellular network. The lab setup for the experiments is illustrated in Figure 2.

In our experiment, the PCBot generates about 3 Mbps traffic while the MobileBot and GameBot generate 0.07 Mbps and 0.25 Mbps respectively. Neither MobileBot nor GameBot have been optimized to gen-

erate the most amount of traffic. Even without optimization, it only takes about 43 phones or 12 gaming consoles to achieve the same effect as a PC. In previous testing, we have demonstrated that a home Internet gateway could be disrupted by low rate DDoS attacks with as low as 5-6 Mbps traffic. Therefore, given the large number of smart phones and gaming consoles, the attacker can easily aggregate the attack to cause devastating effect.

3 Defense

We implemented a method of detecting such an attack in our lab. The detection is based on cflowd records and a commercial network-flow-based detection device. In this sense, it is similar to [8]. ISP carrier-class routers have the capability of creating a type of meta-data for the packets traversing across a router's interfaces. This meta-data is contained in records called cflowd records. The cflowd records are a 5-tuple of information consisting of source and destination ip addresses, source and destination port numbers and protocol. Cflowd records are based on sampling, i.e. the router samples 1 in n packets traversing a cflowd-enabled interface and creates the records based on the sampled packets. We used a 1 to 1 sampling for this experiment.

We configured the flow-based detection device to monitor the simulated consumer IP subnet, as well as the IRC chat port number for the IRC server IP address. We configured alerts based on the chat port number and a packets per second threshold for the consumer IP subnet. The detection device initially detects the IRC server-to-bots communication that consists of two IP packets exchanged when the consumer device is compromised and registers with the IRC server. This is possible since the routers are creating cflowd records for every packet transiting its interfaces and the alert is based on any traffic to the chat port number. Subsequently, it detects the attack itself by triggering on the elevated packets per second rate of traffic destined to the consumer IP range. Once an attack is detected, it can be mitigated by sending the victim's traffic through scrubbers that would filter out the malicious traffic or even by blocking device traffic.

Our experimental configuration only looks for IRC communications. In practice, botnets may tunnel IRC through other protocols like HTTP/HTTPS, or use these protocols directly. Detecting such botnets requires analysis of their traffic pattern and configuring our detection device accordingly.

4 Conclusions and Continuing Work

In this paper, we showed how to create a hybrid botnet using various computing devices including traditional PCs, as well as smartphones and gaming consoles. DDoS attacks implemented using the hybrid botnet and the corresponding network-flow-based detection method were also presented. These experiments have demonstrated that security threats such as botnets are becoming more realistic to emerging intelligent electronic devices. We are building on current work by using network flow data to gather more security-relevant information from the home Internet gateway and other elements near the network edge that can be used to provide a combination of detection, notification, mitigation, and protection against these types of attacks.

References

[1] Devkitpro. http://www.devkitpro.org/.

[2] Host of internet spam groups is cut off. Washington Post, Nov 12, 2008.

[3] Pong2 game. http://www.plaatsoft.nl/wiibrew/pong2/.

[4] Rbot. http://www.f-secure.com/v-descs/rbot.shtml.

[5] Virca IRC MIDLet. http://www.vidarholen.net/contents/virca/.

[6] VLC media player. www.videolan.org/vlc/.

[7] Mcafee threats report: First quarter 2009, 2009. McAfee Avert Labs.

[8] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.